

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

Ю. П. Кондратенко

«__» _____ 2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**СИСТЕМА ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ
ДЛЯ ВПРОВАДЖЕННЯ РОЗУМНОГО БУДИНКУ**

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21610801

Студент _____ Баришніков В. О.

«__» _____ 2022 р.

Консультант _____ Мещанінов О.П.
д.пед.н, проф.

«__» _____ 2022 р.

Миколаїв – 2022

КАЛЕНДАРНИЙ ПЛАН

Виконання магістерської кваліфікаційної роботи

Тема: Система інтелектуального аналізу даних для впровадження розумного будинку

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2021	10.10.2021	
2	Отримання завдання на виконання МКР	19.10.2021	22.10.2021	
3	Складання календарного плану на період виконання МКР	23.10.2021	26.10.2021	
4	Огляд літератури за темою дослідження	27.10.2021	10.11.2021	
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	29.11.2021	18.12.2021	
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	23.12.2021	02.01.2022	
7	Описфасхової частини МКР, зокрема аналіз існуючих різновидностей розпізнавання активностей в реальному часі, реалізація обраних технологій з аналізом отриманих результатів	03.01.2022	25.01.2022	
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2022	30.01.2022	
9	Попередній захист МКР на засіданні комісії кафедри	31.01.2022	31.01.2022	
10	Корегування роботи за результатами попереднього захисту	01.02.2022	03.02.2022	
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	04.02.2022	08.02.2022	
12	Подання МКР рецензенту	11.02.2022	12.02.2022	
13	Рецензування МКР	13.02.2022	14.02.2022	
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.02.2022	17.02.2022	
15	Захист МКР перед екзаменаційною комісією (ЕК)	23.02.2022	24.02.2022	

Розробив студент Баришніков. В. О. _____
(прізвище та ініціали)

(підпис)

Керівник роботи д.пед.н, проф. Мещанінов О.П. _____
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

«24» жовтня 2021 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 607 ЧНУ ім. Петра Могили

Баришнікова Вадима Олександровича

на тему: **“Система інтелектуального аналізу даних для впровадження
розумного будинку”**

Актуальність даного дослідження полягає у тому що на сьогоднішній день можна спостерігати посилений інтерес до ринку розумних будинків зі сторони великих компаній, таких як Google, Samsung, Amazon та інших.

Однак, не зважаючи на ріст ринку, розумні будинки мають певну кількість обмежень серед яких неможливість аналізувати поведінку мешканців будинку.

На даний момент відсутні рішення та підходи які б були в змозі вирішити дану проблему. Саме тому було поставлено задачу розробити програмну систему аналізу активності мешканців будинку. Подібна система може бути використана у декількох напрямках, серед яких автоматизація та охорона здоров'я для моніторингу життя людей з вадами здоров'я і людей похилого віку.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології розумного будинку.

Предметом дослідження є комп'ютерні системи інтелектуального аналізу даних в умовах розумного будинку.

Метою дослідження є розробка програмного забезпечення інтелектуального аналізу даних в умовах розумного будинку, яке має можливості щодо розпізнавання активності у реальному часі з можливістю подальшого використання результатів. У кваліфікаційній роботі представлена розроблена система інтелектуального аналізу даних для впровадження розумного будинку.

У першому розділі розглянуто та проаналізовано особливості проекту CASAS, а також методи сегментації потоку даних сенсорів.

У другому розділі було запропоновано ряд моделей машинного навчання і приведено опис використання ідеї моделювання.

Третій розділ надає засоби розробки, аналіз їх особливостей та переваг, описана загальна схема роботи окремо при завантаженні даних тестування і тренування, а також сценарії роботи користувача з розробленою системою використовуючи діаграму використання.

У четвертому розділі було встановлено технічні вимоги для можливості роботи користувача з розробленою програмною системою і приведений опис вікон програмної системи.

П'ятий розділ пояснює економічне обґрунтування впровадження проекту, де аналізується можливість розробки стартап проекту на базі створеної програмної системи аналізу даних.

Магістерська робота складається зі вступу, п'яти розділів, висновку, переліку посилань з 59 найменувань, додатків, містить 34 рисунків і 16 таблицю. Повний обсяг магістерської роботи складає 60 сторінок.

Ключові слова: розумний будинок, аналіз даних, авторизація, розпізнавання активності, сегментація, машинне навчання, сенсори руху.

ABSTRACT

to the master's qualification work by the student of the group 607 of Petro Mohyla
Black Sea National University

Baryshnikov Vadym

“Data mining system for smart home implementation”

The relevance of this study lies in the fact that today we can observe an increased interest in the smart home market from large companies such as Google, Samsung, Amazon and others. However, despite the growth of the market, smart homes have a certain number of limitations, including the inability to analyze the behavior of residents of the House. At the moment, there are no solutions and approaches that would be able to solve this problem. That is why the task was set to develop a software system for analyzing the activity of residents of the House. Such a system can be used in several areas, including automation and healthcare, to monitor the lives of people with disabilities and the elderly.

Object of research serves computer information systems and smart home technologies.

Subject of research is data mining systems in a smart home.

Purpose is to develop software for data mining in a smart home, which has the ability to recognize activity in real time with the possibility of further use of the results. The qualification work presents a developed data mining system for the implementation of a smart home.

The first section discusses and analyzes the features of the CASAS project, as well as methods for segmenting the flow of sensor data.

The second section proposed a number of machine learning models and described the use of the idea of modeling.

The third section provides development tools, analysis of their features and benefits, describes the general scheme of work separately when downloading test and training data, as well as scenarios of the user with the developed system using the usage diagram.

The fourth section sets out the technical requirements for the user's ability to work with the developed software system and describes the windows of the software system.

The fifth section explains the economic justification of the project implementation, which analyzes the possibility of developing a startup project based on the created software system of data analysis.

The master's thesis consists of an introduction, five chapters, a conclusion, a list of references from 59 titles, appendices, contains 38 figures and 21 tables. The full volume of the master's thesis is 60 pages.

Keywords: smart home, data analysis, authorization, activity recognition, segmentation, machine learning, motion sensors.

ЗМІСТ

1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	9
1.1 Аналіз досліджень	9
1.2 Сегментація даних для аналізу в реальному часі	10
1.3 Методи для аналізу даних в реальному часі	13
Висновки до розділу 1.....	19
2. ПОРІВНЯННЯ МОДЕЛЕЙ КЛАСИФІКАЦІЇ ДАНИХ ПОТОКУ СЕНСОРІВ РОЗУМНОГО БУДИНКУ У РЕАЛЬНОМУ ЧАСІ	20
2.1 Модель випадкового лісу.....	20
2.2 Модель нейромережі прямого поширення	21
2.3 Модель рекурентної нейронної мережі.....	22
2.4 Метод опорних векторів	25
2.5 Особливості формування вектору ознак для обраних моделей і тренування	29
2.6 Аналіз результатів моделювання.....	30
Висновки до розділу 2.....	41
3. РОЗРОБКА СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ РОЗУМНОГО БУДИНКУ.....	34
3.1 Програмні засоби розробки	34
3.2 Загальна схема функціонування системи.....	36
3.3 Сценарії використання системи	38
3.4 Структура програмного забезпечення.....	39
Висновки до розділу 3.....	41
4. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З РОЗРОБЛЕНИМ ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	43
4.1 Технічні вимоги.....	43
4.2 Інтерфейс програмного продукту.....	43
4.3 Робота з аномальними активностями.....	53
Висновки до розділу 4.....	55

5. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ВПРОВАДЖЕННЯ ПРОЕКТУ... 57	
5.1 Опис ідеї проекту	56
5.2 Технологічний аудит проекту.....	57
5.3 Розроблення ринкової стратегії проекту.....	59
Висновок до розділу 5.....	61
6. МЕТОДИЧНИЙ РОЗДІЛ..... 62	
7. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ 91	
ВИСНОВКИ..... 108	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 110	
ДОДАТКИ 116	

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«СИСТЕМА ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ВПРОВАДЖЕННЯ РОЗУМНОГО БУДИНКУ»

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21610801

Студент _____ Баришніков В. О.

«__» _____ 2022 р.

Консультант _____ Мещанінов О.П.

д.пед.н, проф.

«__» _____ 2022 р.

Миколаїв – 2022

ВСТУП

Актуальність дослідження. На сучасному етапі у зв'язку з посиленою інформаційно-технологічною спрямованістю спостерігається й невідступний інтерес до розумних будинків, особливо з боку ринкових компаній та компаній-гігантів, серед яких і Google, Samsung, Amazon тощо. Оскільки такі будинки є пріоритетом стратегічного енергетичного планування та політики безпеки, ринок розумних будинків лише збільшуватиме попит. Але разом зі зростанням у сфері ринку, розумні будинки все ще не можуть гарантувати аналіз поведінку мешканців будинку. Відсутність вирішення такої задачі, як і відсутність ідей спрямованих на уможливлення аналізу активності мешканця будинку зумовлює актуальність даного дослідження. Основним завданням на сьогодні представляється розробка специфічної програмної системи аналізу активності людини у межах смарт-дому. Застосування такої програми дало би змогу поліпшити функціонування одразу кількох сфер використання розумного будинку, а особливо автоматизація та охорона здоров'я для моніторингу життя людей з дефектами здоров'я та людей похилого віку.

Головною особливістю розумних будинків є те що в них використовуються пристрої які здатні зберігати та обмінюватися інформацією. Ця інформація може бути використана для аналізу активностей мешканців будинку [49].

Для аналізу даних можуть бути використані різні джерела інформації, наприклад камери спостереження або бінарні сенсори руху. Але в рамках цієї роботи були вирішено аналізувати дані з бінарних сенсорів руху. Основною причиною такого вибору є низька вартість сенсорів та економія ресурсів за рахунок того що дані зберігаються у звичайних текстових файлах. Також вагомим плюсом є те що дані цих сенсорів не є настільки критично конфіденційними для мешканців будинку.

На сьогоднішній день методики аналізу активностей представлені в 2 групах. В першій групі методики базуються на знаннях а в другій на даних.

Ті підходи, які базуються на знаннях, використовують євристики, а також попередні знання предметної області для моделювання активностей. По цій причині таким моделям не потрібно мати великі набори даних, але вони є дещо статичними до конкретно заданих характеристик розумного будинку [46].

З іншої сторони, ті підходи, які базуються на даних, використовують моделі машинного навчання, наприклад метод опорних векторів, нейронні мережі та інші. За цим підходом, необхідно надати набір даних для тренування, але ці набори даних, можна використовувати у різних за характеристиками розумних будинках.

Також ці підходи можна поділити на ті, які використовують навчання з вчителем та навчання без вчителя. Для навчання без вчителя використовується різні методи кластеризації. Зазвичай, на виході цих моделей є шаблони, які можуть бути наявні в даних або існуючі в даних аномалії [5].

З іншого боку, для навчання з вчителем використовуються анотовані активності. Тобто, на виході ми отримуємо множину класифікованих активностей. Існують варіанти, коли використовується лише одна модель класифікацій або гібридні підходи, коли підбираються різні моделі класифікацій та кластиризації, в такому випадку результат кластеризації буде переданий на вхід моделям класифікацій [39–10].

Одною з переваг системи аналізу сенсорів є те що вона може функціонувати у реальному часі. Також на ринку не представлено систем, які б могли надати користувачу можливість класифікації даних для їх використання.

Розроблену програмну систему можна використовувати для моніторингу активностей людей похилого віку та людей з вадами. Таким чином, за необхідності, медичні або соціальні працівники зможуть вчасно реагувати, виходячи з моніторингу активностей. Саме тому, для таких випадків, важливим є наявність можливості виділення аномальних активностей [29].

Такоже ця система надасть можливість автоматизації. Завдяки наявним класифікованим активностям, розробники будуть мати можливість зробити більш персоналізовані налаштування. Наприклад буде можливим створення правила для виключення світла у випадку сну.

Виходячи з цього **метою** даної роботи є розробка програмного забезпечення для інтелектуального аналізу даних в умовах розумного будинку,

Дана система має можливості щодо розпізнавання активностей у реальному часі, та має можливості подальшого використання результатів.

Для досягнення мети були поставлені наступні **завдання**:

- дослідити існуючі підходи розпізнавання активностей у реальному часі;
- удосконалити метод який буде використовуватися для розпізнавання активностей у реальному часі;
- розробити програмну систему, яка матиме змогу використовувати розроблену модель, та на основі цього візуалізувати результати розпізнавання і аналізу аномалій та активностей.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології розумного будинка.

Предметом дослідження є комп'ютерні системи інтелектуального аналізу даних в умовах розумного будинку.

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результатами магістерської роботи є:

- покращення методу розпізнавання активностей, використовуючи дані отримані з бінарних сенсорів руху у реальному часі та використавши модель випадкового лісу як класифікатора та модель Латентного Розміщення Діріхдлу для того щоб включити в сегмент інформацію щодо розподілу активностей, що надасть зростання точності класифікації;
- розробка моделі класифікації щодо даних потоку сенсорів та можливість формування вектору ознак;
- відображає результати та аналіз аномальних активностей.

- розробка програмної системи яка використовує розроблену модель;

Магістерська робота складається зі вступу, чотирьох розділів, висновку, 59 59 посилань, додатків, містить 34 рисунків і 21 таблиці.

1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

З метою розробити оптимальну модель для класифікації активності у реальному часі варто розглянути існуючі дослідження та проаналізувати отримані результати. Ключовим завданням представляється вибір набору даних для здійснення тренування моделі, оскільки так валідація відбуватиметься на тому ж наборі даних. Дані повинні максимально нагадувати ті, що надходять у ході застосування розробленої програмної системи. Так був вибраний проект Вашингтонського університету центру досліджень адаптивних систем, що обумовлюється наявністю значної кількості наборів даних, які найбільше всього підходять для тренування та валідації моделей класифікації.

1.1 Аналіз досліджень

Проблематика аналізу функціонування людини в межах розумного будинку була запропонована проектом Вашингтонського університету центру досліджень адаптивних систем CASAS (Center for Advanced Studies in Adaptive Systems at Washington State University). Результатом проекту стало виокремлення поведінкових шаблонів.

За період проживання мешканців будинку було зібрано необхідні дані. Переважна більшість наявних у будинку активностей розмічено згідно з активаціями сенсорів, де у наборі поставлена та чи інша активність. Такі активності можуть бути розміщені у довільному порядку. Існують фактори, що можуть відрізнити активності між собою, серед яких імплементація розмітки, спеціальні дослідження, планування будинку і розташування сенсорів всередині нього, обсяг сенсорів і анотованих активностей, а також число мешканців.

Можемо простежити, що CASAS включає апаратні компоненти (сенсори та актуатори), а архітектура послуговується мережею Zigbee, де та, у свою чергу, безпосередньо зв'язується зі встановленими пристроями. Рівнем

проміжного програмного середовища керує менеджер. Менеджером надаються канали, за допомогою чого компонентні мости відправляють та отримують повідомлення. Мітки з часовим інтервалом додаються також менеджером, який з'єднує їх з виникаючими подіями і присвоює UUID. Кожен компонент архітектури CASAS обмінюється повідомленнями з цим менеджером через спеціальний міст XMPP.

Інформація у наборах даних – це послідовно зібрані показники сенсорів, у якому кожен ряд передає дані про час активації сенсору зі специфічним унікальним ідентифікатором та значенням, як і ановану активність (Рис. 1.1).

2010-11-04	05:40:51.303739	M004	ON	Bed_to_Toilet	begin
2010-11-04	05:40:52.342105	M005	OFF		
2010-11-04	05:40:57.176409	M007	OFF		
2010-11-04	05:40:57.941486	M004	OFF		
2010-11-04	05:43:24.021475	M004	ON		
2010-11-04	05:43:26.273181	M004	OFF		
2010-11-04	05:43:26.345503	M007	ON		
2010-11-04	05:43:26.793102	M004	ON		
2010-11-04	05:43:27.195347	M007	OFF		
2010-11-04	05:43:27.787437	M007	ON		
2010-11-04	05:43:29.711796	M005	ON		
2010-11-04	05:43:30.279021	M004	OFF	Bed_to_Toilet	end
2010-11-04	05:43:45.7324	M003	ON	sleeping	begin
2010-11-04	05:43:52.044085	M003	OFF		
2010-11-04	05:43:53.185335	M002	ON		
2010-11-04	05:43:53.253809	M003	ON		
2010-11-04	05:43:59.493281	M002	OFF		
2010-11-04	05:44:04.048766	M003	OFF		
2010-11-04	05:44:06.14204	M003	ON		
2010-11-04	05:44:11.229146	M003	OFF		

Рис. 1.1. Приклад вхідних даних

У кожному наборі даних знаходиться план розташування описаних сенсорів у будинку.

1.2 Сегментація даних для аналізу в реальному часі

Вимогою даної роботи є можливість системи аналізувати потік даних у реальному часі. Щоб уможливити цей процес та виробити класифікацію потік

вхідних даних з сенсорів ділиться сегменти, а згодом розроблена модель аналізується сенсорами. Так, навчальну вибірку становитимуть сегменти, де кожному відповідає певна активність.

Поділ на сегментації: за активностями, за часом і за сенсорами [3]. Види сегментації представлені на рисунку 1.3.

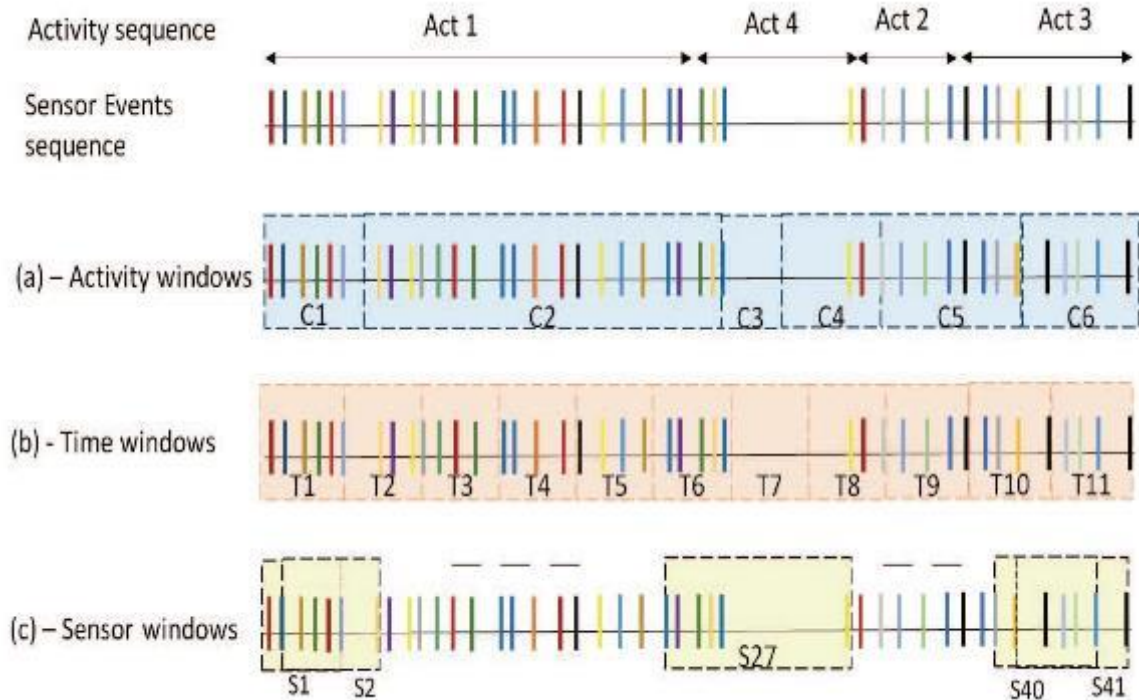


Рис. 1.3. Види сегментації даних потоку сенсорів

Під час сегментації за активностями вхідні сенсори на першому етапі розподіляються на вікна різної довжини. Було заплановано: кожне вікно репрезентує певну активність. Наступним етапом вікно передається на вхід до прогнозуючої моделі класифікації. Однак проблемою постає той факт, що кілька активностей можуть займати місце в одному вікні, тоді, як наслідок, такий спосіб прогнозує й більшу кількість активацій сенсорів для подальшої класифікації.

У ході сегментації за критерієм часу потік сенсорів розподіляється на вікна однакової часової довжини. Перепоною такого підходу виступає відсутність достатніх даних, щоб класифікувати активність, адже залежно від вибору довжини вікна кожне вікно може не містити достатньої інформації для

здійснення класифікації активності, і навпаки, включати дані, які відповідають декільком активностям.

За сегментації за сенсорами спостерігаємо поділ потоку сенсорів на вікна з рівномірною кількістю активацій сенсорів. Доцільно апелювати до класифікації останнього сенсору в вікні, визначниками контексту слугують саме минулі сенсори. У задіяному методі спостерігаються й недоліки: вікно може містити в собі активації сенсорів, між якими великі проміжки часу. За умови присутності в будинку декількох мешканців, кожне вікно може потенційно включати дані щодо декількох паралельних активностей. Вибір вказаного методу обумовлений здійсненням класифікації без очікування певної кількості сенсорів. Помітно, що класифікація реалізується під час надходження кожного сенсору. Обрана ширина вікна визначає кількість активацій сенсорів в сегменті.

Оскільки система має класифікувати кожну активацію сенсору на конкретну активність у реальному часі – сегментація дозволяє використати певну кількість минулої інформації для здійснення класифікації останнього сенсору. Більш детально цей процес зображений на рисунку 1.4.

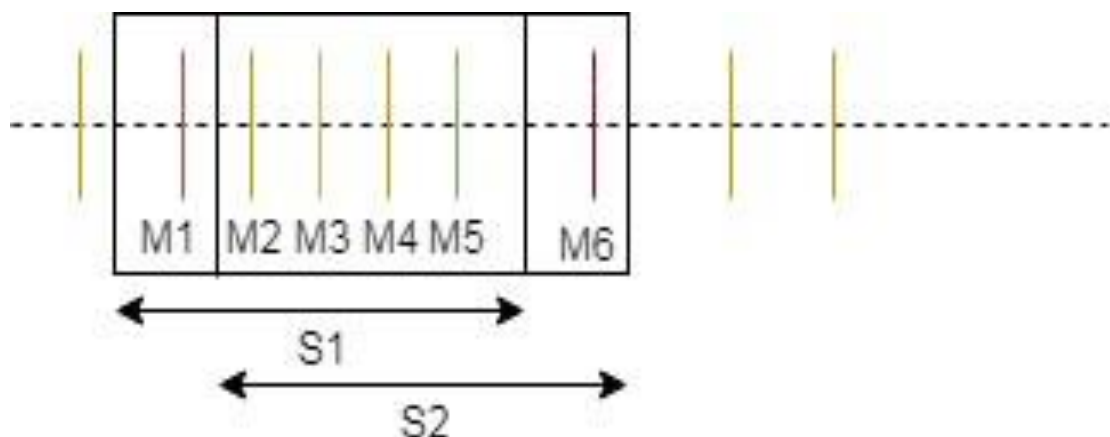


Рис. 1.4. Сегментація за сенсорами

Нехай ми маємо на вході активацію сенсора M5 і маємо довжину вікна 5. На базі цього формується сегмент S1 до якого входять активації M1, M2, M3, M4. На наступному кроці коли на вхід потрапляє сенсор M6 – відповідно

формується сегмент S2 до якого входять активації M2, M3, M4, M5. Надалі з сформованих сегментів формується вектор ознак.

1.3 Методи для аналізу даних в реальному часі

Під час дослідження було здійснено відбір спектру робіт сфокусованих на аналізі у реальному часі. На наступному етапі заплановано порівняння точності розробленої моделі з тими результатами, яке пропонує дана робота.

Одне з сучасних досліджень висуває ідею аналізу різнобічних підходів до формування вектору ознак. Основний підхід вміщує в кожний сегмент дані про час активацій першого та останнього сенсору у вікні, тривалість та кількість активацій кожного з сенсорів. Решта підходів репрезентують видозміни базового. Наприклад, один підхід презентує зважування частот активацій сенсорів базуючись на часовій дистанції до останньої активації сенсору.

Інший підхід пропонує залучити в сегмент інформацію про минуле вікно. Як результат, відбувається тренування двох моделей, одна з яких тренується на сегментах безвідносно до інформацію про минуле вікно, а інша – перетворює результат класифікації першої моделі на якісно нову ознаку.

Важливим постає дослідження й аналіз сегментації за часом з різними довжинами вікон, зокрема в діапазоні від п'яти секунд до хвилини. Нові дані може дати й метод динамічного вибору довжини вікна шляхом отримання статистичної інформації представлений в даних. У такий спосіб може бути вирахована необхідна довжина вікна, беручи до уваги інформацію про довжини вікон активностей і розподілу сенсорів по цим активностям.

Метод опорних векторів якості дозволяє здійснювати класифікації. Як результат, найдостовірніші дані отримано за допомогою поєднання підходів з матрицею взаємної інформації та інформації про минулу активність.

Існує й відмінний погляд на застосування методів, а саме додавати до вектору ознак ідентифікатора останнього сенсора вікна і задіяння матриці взаємної інформації, що буде брати до уваги лише імовірність активації двох сенсорів в рамках однієї активності.

Для зважування частот у векторі ознак може бути використана інформація у матриці. Дослідження також апелює до методу опорних векторів, адже вище наведені підходи не дали достовірної класифікації. Причина негативного результату пов'язана з наборами даних CASAS, у яких зберігаються лише частково анотовані активності. Відомо, на потоці даних сенсорів активації сенсорів не відповідають якійсь конкретній єдиній активності. Що стосується неанотованих активацій, то такі активації потенційно можуть стосуватися й переміщення мешканця по будинку, або непоміченій дослідженням активності. Закономірно, нерозмічені активації сенсорів маркуються окремим класом, далі відбувається сегментування та подача на вхід до класифікатора, що типово, як і для інших класів. Простежується незбалансованість навчальної вибірки, що становить проблему для дослідження. Візуально можемо прослідкувати описаний розподіл активацій сенсорів за належними класами в одному з наборів даних CASAS на представленому рисунку 1.6.

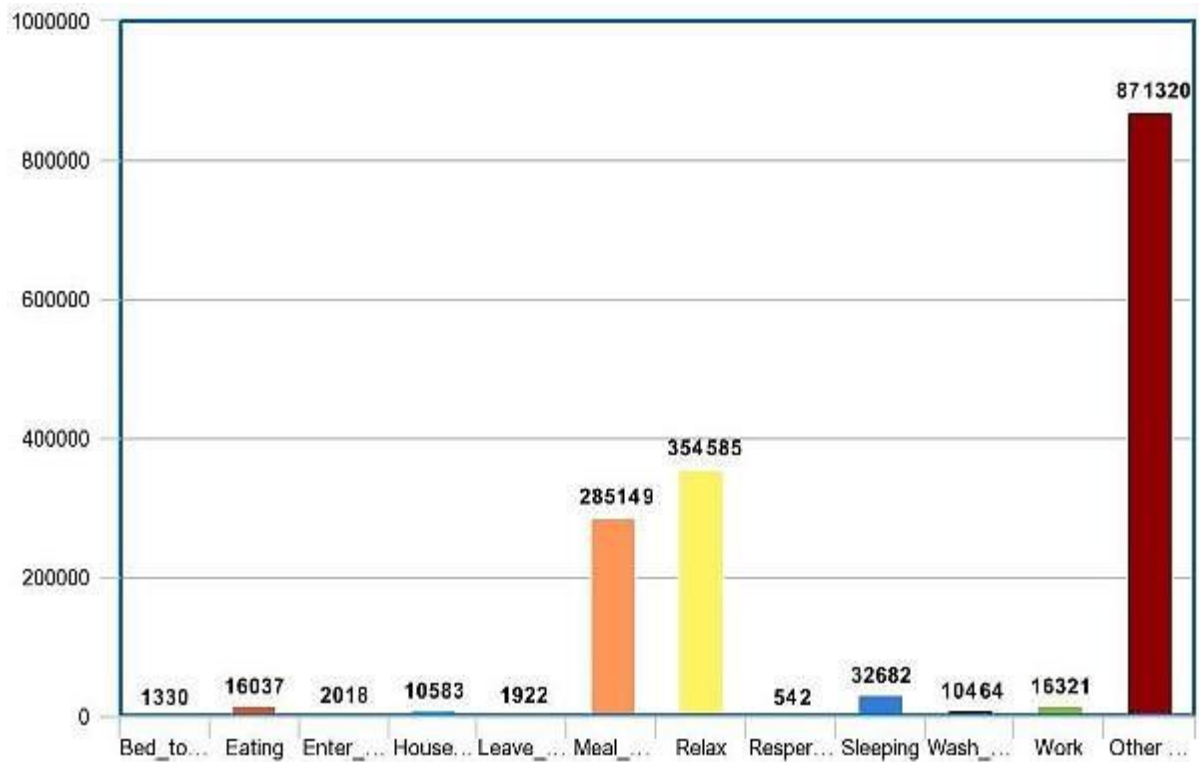


Рис. 1.6. Розподіл активацій сенсорів за класами

Наслідком вище заявленої незбалансованості є збільшення імовірності не достовірної класифікації, де якась з активностей може залишитися не відслідкованою та не розміченою. Цей феномен демонструє матриця помилок дослідження [18].

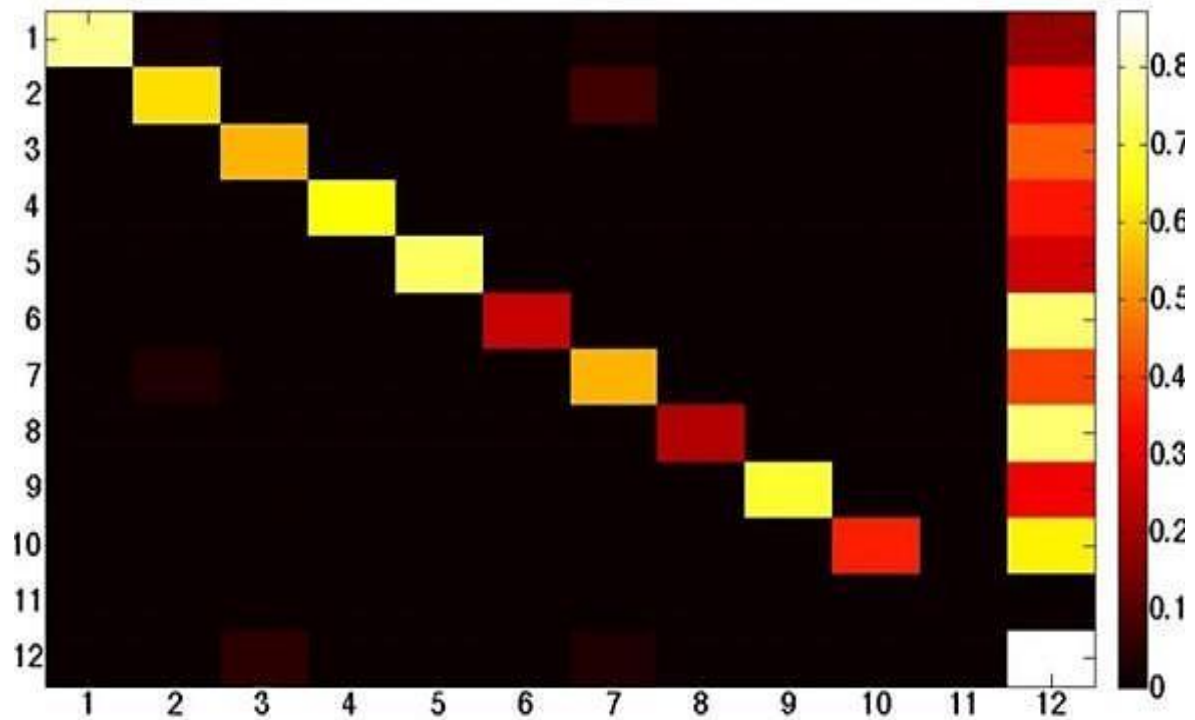


Рис. 1.7. Матриця помилок дослідження

Схематичність матриці орієнтована на окреслення помилок, де вертикальні позначення вказують на правильні мітки класів, тоді як горизонтальні – класифіковані. Можемо зробити висновок про значний відсоток класів, яких зараховано до іншої активності.

У деяких дослідженнях пропонується застосування методів формування вектору ознак або вище перелічені методи витворення класифікації. Якщо взяти до уваги метод найближчих сусідів з вкрапленням модифікацій, то у такому разі присвоєння класу з вхідним вектором базується на зважування класів, до яких прилягають найближчі вектори. Доцільність такого методу обумовлюється уможливленням пошуку по елементам навчальної вибірки без спеціального навчання. Крім того, наступні модифікації були задіяні у ході роботи

- оперування векторами наявних класів, щоб резолювати питання незбалансованості навчальної вибірки;
- метод під назвою «гаусівська куля» передбачає тренування, однак може бути ефективним у вирішенні проблеми незбалансованості;
- видозміна наявних методів для аналізу активності, пошуку проблем.

Використані методи однак лише підтвердили інтенсивність на шляху до точності класифікації, яка може бути досягнута через відслідковування потенційних похибок класифікації через непомічені активності.

Поліпшення достовірності класифікації (на 5%) допомогли методи включення станів сенсорів до вектору ознак та включення декількох останніх ідентифікаторів сенсорів, а також метод опорних векторів.

Важливою є ідея побудови матриці найчастотніших сенсорів кожної активності. Сформувавши ряд статистичних рис, які складають середнє, медіана, стандартне відхилення тощо допомагають детектувати доцільність сенсорів на основі зросту інформації. У цьому випадку оптимально для формування класифікації було запельовано до методу опорних векторів, що, як правило, вимагає окремого тренування для кожного класу.

Схематично відтворити отримані кращі дані з описом їх характеристик можна наступним чином у вигляді таблиці. Сформувати точну оцінку допомогли такі метрики, як точність за класами й метрика F1. Представлені метрики ефективно вирішують питання мультикласової класифікації та незбалансованості отриманих даних. Достовірність за класами можна розрахувати у таких спосіб:

$$\text{Загальна Точність} = \sum_{j=1}^A \frac{TPa_j}{Na_j} \quad (1.4),$$

де A – кількість класів, TPa_j – кількість вірно класифікованих сегментів для класу a_j , а Na_j – загальна кількість сегментів цього класу.

Особливість метрики F1 полягає у тому, що вона враховує точність та повноту результуючого класу. Доцільність представляється частиною належним чином прокласифікованих екземплярів у класі, а повнота – це частина належних класифікацій щодо існуючих в наборі даних екземплярів. Умовно позначемо точність як a його – повнота, відповідно метрика F1 для даної задачі визначається у такий спосіб:

$$F1 \text{ метрика} = \frac{2}{A} \sum_{j=1}^A \frac{Pa_j * Ra_j}{(Pa_j + Ra_j)} \quad (1.5).$$

За результатами використання двох метрик приведемо дані точності класифікацій досліджень без іншого класу у таблиці 1.1.

Таблиця 1.1

Представлені результати досліджень без іншого класу

Дослідження	[40]	[41]	[37]
Точність	87.55%	86.76%	85%
Метрика F1	69.24%	62.23%	82%

Представлено 3 публікації так як не всі тестували точність не враховуючи інший клас. У таблиці 1.2 представлено результати класифікаторів враховуючи класифікацію іншого класу.

Таблиця 1.2

Представлені результати досліджень з іншим класом

Дослідження	[18]	[40]	[41]	[6]
Точність	61%	63%	60%	59%
Метрика F1	48%	47%	47%	58%

Порівняння наведених таблиць показує, що класифікація іншої активності суттєво знижує достовірність наявних класифікаторів.

Висновки до розділу 1

Перший розділ присвячено наступним питанням:

Окреслено значення проекту CASAS для імплементації системи розумного будинку з розміченими активностями. Здійснено вибірку даних у реальних умовах та представлено у вигляді потоку даних сенсорів. Набір даних представлено у максимально наближеному вигляді, імітуючи реально надходжувальні дані, що допомагає застосовувати їх під час тренування чи валідація моделей.

Проаналізовано методи сегментації потоку даних сенсорів, де ефективність щодо вирішення поставленої задачі виявлена у методу сегментації за сенсорами, адже уможлиблює класифікацію без очікування даних з наступних сенсорів.

Зазначені методи є продуктивними у межах різних дотичних напрацюваннях. Закономірно, найпродуктивнішим і найбільш регулярним є застосування методу опорних векторів з формуванням різних векторів ознак. Важливо використовувати різні моделі машинного навчання на різних векторах ознак.

Таким чином, включення іншого класу до класифікації впливає на точність (достовірність) у сторону її зниження до значення, якого не достатньо для підходів описаних вище та використовуваних у реальних умовах.

2. ПОРІВНЯННЯ МОДЕЛЕЙ КЛАСИФІКАЦІЇ ДАНИХ ПОТОКУ СЕНСОРІВ РОЗУМНОГО БУДИНКУ У РЕАЛЬНОМУ ЧАСІ

Виходячи з підходів які були проаналізовані в минулому розділі потрібно розробити власний підхід до класифікацій. Важливо враховувати при виборі моделі машинного навчання що для класифікації важливим є не тільки сама модель а й вектор ознак що був сформований для неї. Для того щоб досягти максимальних результатів і вирішити всі задачі, було вирішено вибрати декілька моделей машинного навчання та порівняти їхню точність.

Необхідно порівняти точність результатів моделювання з представленими раніше дослідженнями. Під час аналізу набору даних стає зрозуміло, що тут має місце незбалансованість даних. Саме тому вибрані нами моделі не повинні мати таких недоліків.

2.1 Модель випадкового лісу

Модель випадкового лісу поєднує в собі певну кількість дерев рішень, кожне таке дерево проводить тренування на підвиборці з даних. Але дані які потрапляють на вхід до дерев можуть повторюватись. Для побудови таких дерев використовується алгоритм CART. За цим алгоритмом будується бінарне дерево, та при побудові на кожному наступному кроці правило в вузлі ділить множину на дві частини. Перша частина - яка відповідає правилу в вузлі, друга - не відповідає. Це правило створюється при спробі поділу набору даних на групи, на базі значення певної ознаки.

Під час спроби побудови використовується критерій Джині. Наприклад, якщо набір даних A має дані n класів, тоді індекс Джині можна визначити таким чином

$$G(A) = 1 - \sum_{k=1}^n p_k^2$$

(2.1) ,

де p_k - відносна частота класу k в A .

Найкращим можна вважати таке розбиття яке мінімізує даний критерій.

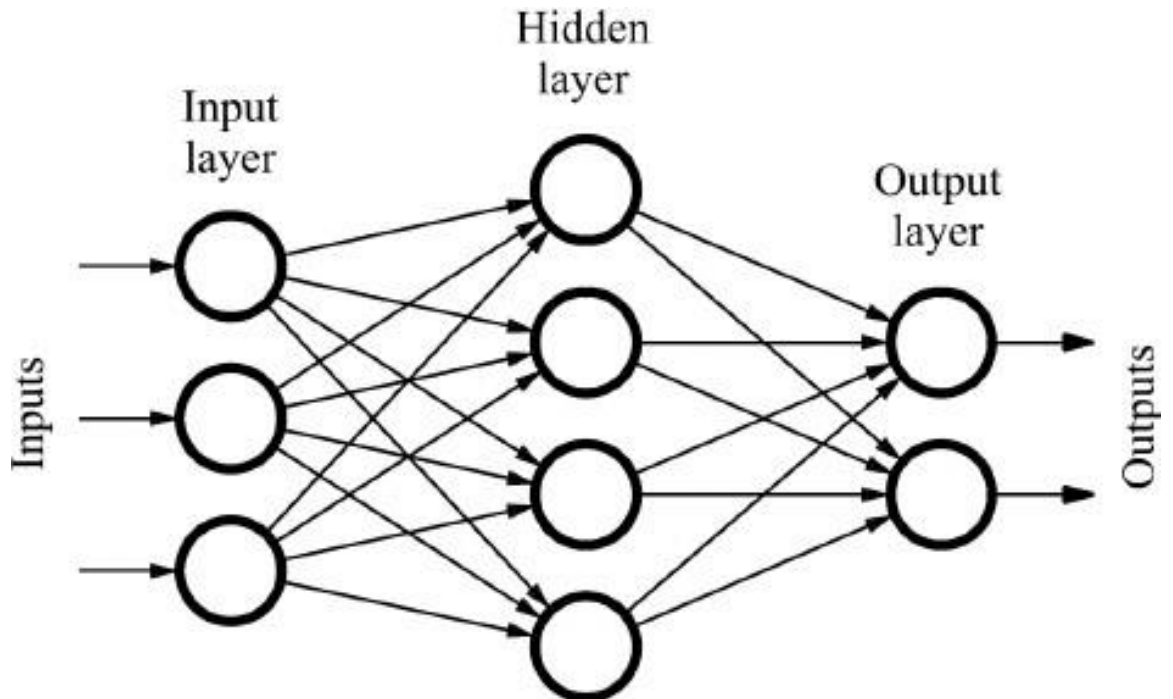
Як тільки навчання дерев завершено, класифікація буде здійснюватись через голосування кожного дерева. Але більшість дерев рішень проголосує за результативний клас. Кількість дерев рішень, зазвичай обирається таким, яке дасть кращий результат на даній валідації.

Найбільшою перевагою цього методу є можливість роботи з великим числом ознак. При цьому, ці ознаки можуть мати різний тип та можливість паралельного навчання, що зменшує необхідний час на тренування. Також ця модель, може працювати і з неточними даними, що є дуже важливим, виходячи з того що існуючі набори даних для даної задачі, найчастіше містять неточності.

2.2 Модель нейронної мережі прямого поширення

Зростання кількості використань моделей глибиного навчання відбулося тому що розробка комплексних нейронних мереж, надає можливість описати більше комплексних залежностей, які існують в даних. Тому саме завдяки таким моделям почали вирішуватись проблеми які раніше не можна було вирішити.

Модель нейронної мережі прямого поширення зображена на рисунку 2.1. Таким чином, звичайна нейронна мережа складається з вхідного слою, деякої кількості скрити слоїв і вихідного слою. При цьому кожен шар має свою певну кількість комірок.



Рисунку 2.1. Модель нейронної мережі прямого поширення

Для кожної комірки буде розраховуватися за:

$$f \left(w_0 + \sum_{i=1}^n x_i w_i \right)$$

(2.2) ,

де w_i – вага, x_i – вхідне значення вектору. Функція f це функція активації.

2.3 Модель рекурентної нейронної мережі

Виходячи з того що набір даних - це потік сенсорів, то тут важливу роль може відіграти послідовність активації сенсорів. Враховуючи що набір даних являє собою потік сенсорів – важливу роль може грати саме послідовність активації сенсорів. А рекурентна нейрона мережа може врахувати цю послідовність.

Саме рекурентні нейронні мережі являються тими мережами, яких використовують для аналізу потокової інформації. Наприклад їх

використовують для розпізнавання мовлення та аналізу тексту, тощо. Вони характеризуються можливістю зберігати внутрішній стан і це надає можливість обробки потокової інформації.

Зазвичай рекурентна нейронна мережа має проблеми зникаючого градієнту, тоді коли градієнтні похибки зникають доволі швидко при навчанні. Тому було запропоновано модель довгої короткочасної пам'яті яка б могла вирішити цю проблему.

Ця модель додає вентиля зберігання, забування та виходу, які контролюють те, яка частина вагів зберігаються чи оновлюються під час навчання мережі. Тому, розробка цієї моделі надала можливість аналізу більш довготривалої залежності в даних. Модель мережі зображена на рисунку 2.3

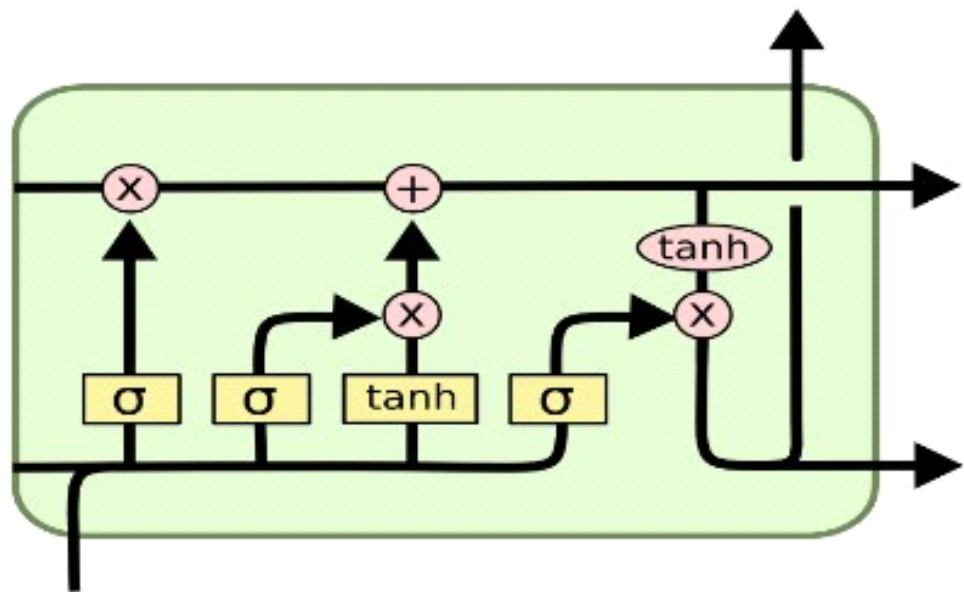


Рис. 2.3. Модель довгострокової нейронної мережі

Модель визначається наступними рівняннями:

$$\begin{aligned}
 i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\
 f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\
 o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\
 C_t &= \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \\
 \tilde{C}_t &= \tanh(x_t U^g + h_{t-1} W^g) \\
 h_t &= \tanh(C_t) * o_t
 \end{aligned}$$

де U – в цьому випадку матриця вагів, між поточним шаром і входом.

i_t , f_t , o_t – вхідний вентиль, вентиль забування та вихідний вентиль. При цьому кожен з вентилів має свою власну матрицю вагів W . Наприклад, вхідний вентиль визначає, кількість інформації яка зберігається в даній комірці. З іншого боку, вентиль забування визначає, часткове зберігання стану попередньої комірки. Вихідний вентиль визначає, частину внутрішнього стану самих комірок яка передається у інші шари чи на інший крок. До цих вентилів застосовується сигмоїдальна функція, і їх виходом є число від 0 до 1. Сигмоїдальна функція має такий вигляд:

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \quad (2.5)$$

На основі тих що становлять новий та минулий стан комірки, вираховується значення. На базі тих, які становлять новий стан комірки та минулий вираховується значення. Вихідний стан h розраховується на основі перемноження розрахованого, до якого має застосовуватися гіперболічна функція і вихідного вентиля.

Розроблена архітектура має наступний вигляд (Рис. 2.4).

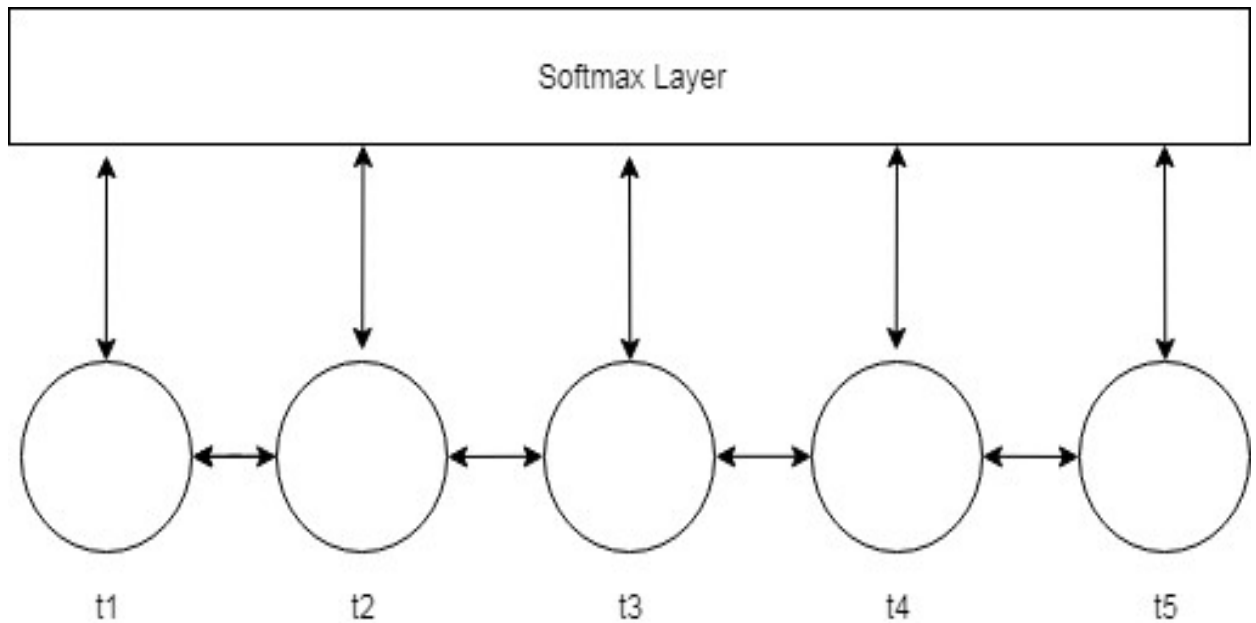


Рис. 2.4. Використана модель довгострокової мережі

Таким чином для здійснення класифікації виходу з комірок довгої короткочасної пам'яті має бути переданий надалі в, і розрахований на основі softmax шару, для можливості багатокласової класифікації.

2.4 Метод опорних векторів

Головна суть цього методу класифікації полягає у пошуку найбільш оптимальної гіперплощини для поділу класів. Цей метод будує функцію класифікації F , яка визначається таким чином:

$$F(x) = \text{sign}(\langle w, x \rangle + b)$$

(2.6),

де x – елемент навчальної вибірки, b – допоміжний параметр, w – нормальний вектор до поділяючої площини

Таким чином, ті варіанти, для яких $F(x) = 1$, відносяться до одного класу, а для яких $F(x) = -1$ відносяться до іншого.

Специфіка цього методу зображена на рисунку 2.5.

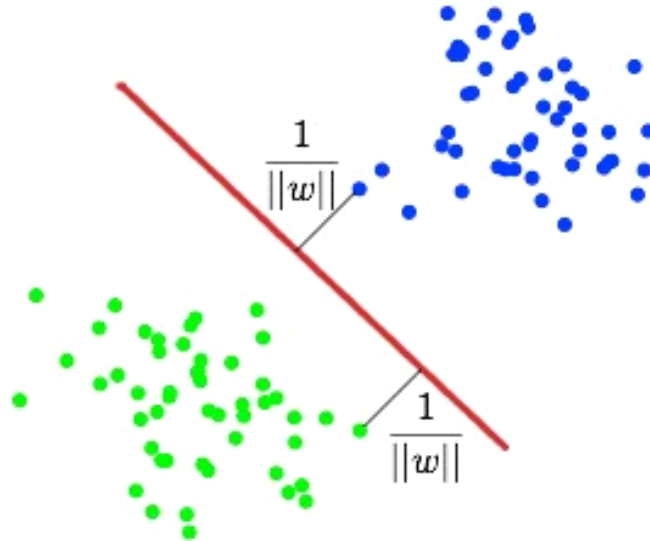


Рис. 2.5. Пошук гіперплощини у методі опорних векторів

Таким чином, потрібно обрати такі значення b та w , при яких, відстань до кожного з класів буде максимальна. Ця відстань дорівнює $1/||w||$. Таким чином,

проблему максимізації $1/||w||$ можна позначити як проблему мінімізації $||w^2||$ [36].

Так задачу:

$$\begin{cases} \arg \min_{w,b} ||w||^2, \\ y_i(\langle w, x \rangle + b) \geq 1, \quad i = 1, \dots, m. \end{cases} \quad (2.7),$$

далі вирішується метод невизначених множників Лагранжа. Якщо врахувати факт, що досить часто зустрічається проблема, тому що класи не можна поділити однією лінією. Тому використовується функція для відображення навчальної вибірки у простір більшої розмірності, таким чином, щоб цю вибірку можна було б розділити лінією.

Приклад даних, які не розділяються лінією зображений на рисунку 2.6.

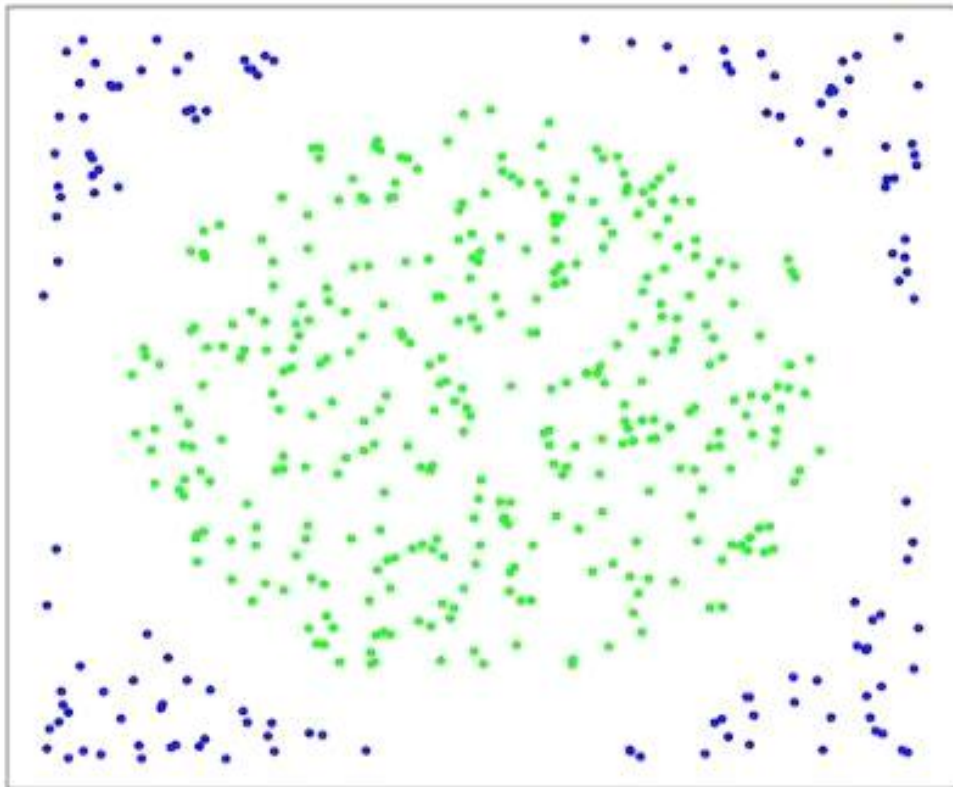


Рис. 2.6. Проблема нероздільності даних лінією

Саме цю функцію називають ядром. В показаних дослідженнях було застосовано радіально-базисне ядро:

$$K(x, x') = e^{-\left(\frac{\|x-x'\|}{2\sigma}\right)^2}$$

(2.8) ,

де σ – це вільний параметр, а x та x' – вхідні вектори.

Потрібно підкреслити, що цей метод розрахований лише на бінарну класифікацію. Для того щоб мати можливість здійснювати багато класової класифікації, цей класифікатор тренують стільки раз, скільки й класів, кожний раз виконуючи бінарну класифікацію поточного класу від усіх інших. Якщо враховувати величину вхідних даних, та достатньо велику кількість активностей, то тренування даного методу потребує велику кількість часу.

Наприклад, моделювання тем залежить від розподілу слів у документах. Приклад такого підходу зображений на рисунку 2.7. Ці моделі можна використовувати не тільки для аналізу тексту, а також в багатьох інших областях, де може бути необхідним такий аналіз.

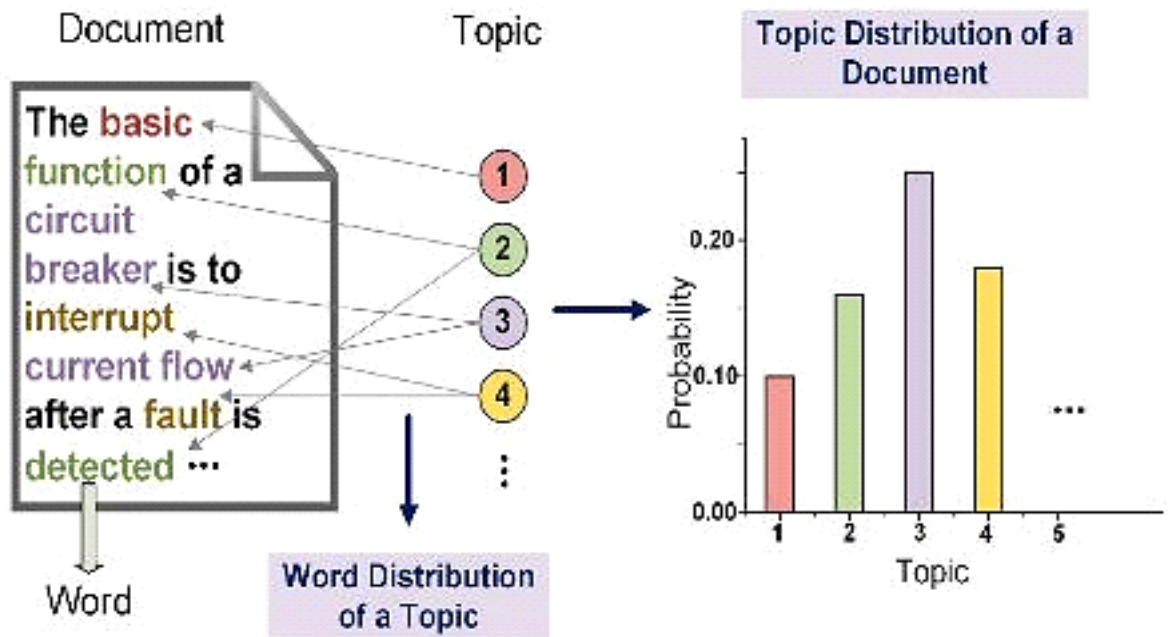


Рис. 2.7. Приклад моделювання теми

Однією із таких моделей є Латентне розміщення Діріхле. Дана модель покращує результати інших схожих моделей. Це відбувається за рахунок стійкості до проблеми перенавчання і можливість додавати нові документи в модель. Якщо кожному документу є можливість поставити у відповідність певний розподіл тем, то в цій моделі вважається, що це розподіл Діріхле.

Виходячи з того, що в деякий момент часу сегмент обмежений в інформації, то є важливим принести частину початкової інформації з навчальної вибірки. Тому, якщо вважати, що документ це вікно активності, а ознаки цього вікна це слова, то сам розподіл активностей є темами, тоді можна використовувати дану модель для включення в кожний сегмент інформації про імовірнісний розподіл активностей на основі ознак в цьому сегменті.

Таким чином, в цьому випадку наявна можливість розробити модель класифікації, використати розподіл ймовірностей активностей у певному вікні. Виходячи з цього, найбільш підходящою можна вважати активність, яка максимізує умовну імовірність вектору ознак у певному вікні:

$$P(y_i = A | w_i) = P(A | F_i^*) = P(A)P(F_i^* | A) \quad (2.9),$$

де A це розподіл активностей, F_i^* – ознаки вікна активності, w_i – сегмент. Для того щоб була можливість створення такого класифікатора, необхідно у кожному вікні активності створити вектори ознак, що включає в себе вектор частот активацій сенсорів та час. В цьому випадку для такої моделі час перетворюється у 24-мірний вектор, який має значення 1 у місцях які відповідають часу тривалості активності. Виходом для моделі латентного розміщення Діріхле є розподіли θ . Через те що дана модель є моделлю без вчителя, її виходом є розподіл власних тем θ^* . Необхідно обрати функцію відображення тем моделі і справжніх активностей:

2.5. Особливості формування вектору ознак для обраних моделей і тренування

Для моделей методу опорного векторів, нейронної мережі прямого поширення та випадкового лісу були обрані наступні ознаки: час початку і кінця, частота активації сенсорів, часова тривалість сегменту, ідентифікатори двох останніх сенсорів і ознаки отримані при використанні моделі Діріхле.

Таким чином врахування двох сенсорів дає значно більше корисної інформації про вікно моделі. Частоти активацій були зважені з використанням матриці взаємної інформації яка була визначена у формулі 1.3. Цей метод зважування частот можна вважати більш доцільним тому що розраховується в рамках вікна активності. Для того щоб мати можливість використання дати у векторі ознаках, сама дата визначається як нормалізована сума часу, хвилин та секунд.

В свою чергу для рекрутерної нейронної мережі на кожному кроці використовується інформація щодо індифікаторів сенсору, значення сенсору та час активації.

2.6. Аналіз результатів моделювання

Було зроблено спробу використати підхід стекінгу, який полягає у використанні декількох класифікаторів для здійснення класифікації. Ідея цього підходу зображена на рисунку 2.10. в цьому випадку, використовується перехресна перевірка, головною ідеєю якої є поділ вибірки на підвибірки та в подальшому обучити класифікатор на одній частині і перевірити на іншій.

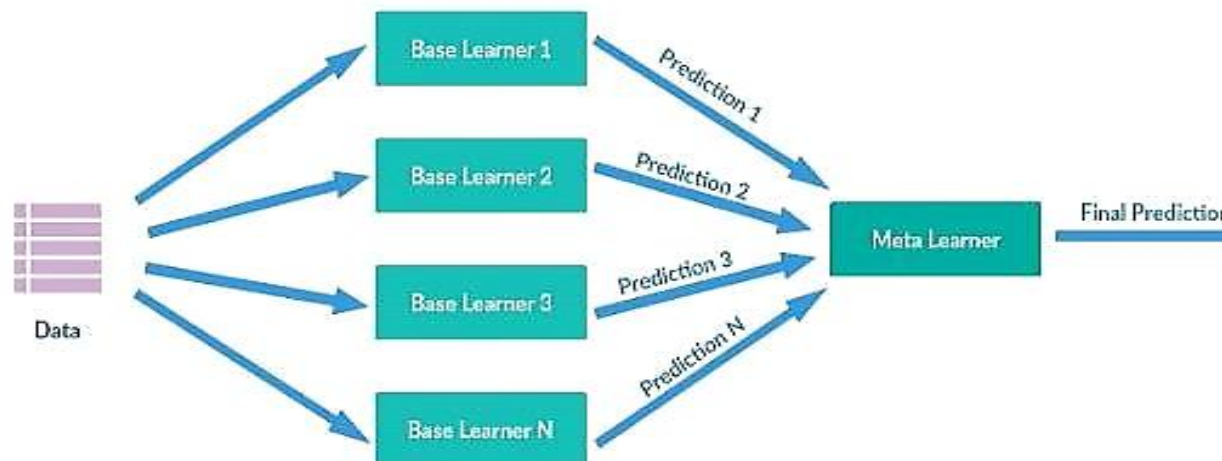


Рис. 2.10. Стекінг класифікаторів

Таким чином створюється нова навчальна вибірка, що складається з результатів класифікації підвиборок на яких проходила перевірка. За допомогою цієї вибірки навчається мета-модель, яка і стає моделю класифікацій. Але в рамках дослідження було зроблено висновки що цей підход не є достатньо точним для використання.

Крім того, була спроба для тренування моделі використати ймовірність класифікацій. Для цього, тренування моделі класифікацій відбувалося без іншої активності. Також для кожної було підраховано важливе значення впевненості класифікацій. В тому випадку, якщо впевненість класифікацій є нижчою за різницю середнього значення та обраної константи, то до сегменту буде присвоєний інший клас.

Один із підходів для тренування моделі, який був використаний це розподіл ймовірностей класифікацій. Таким чином тренування моделі класифікацій відбувалося без іншої активності і для кожної з активностей було розраховано середнє значення впевненості класифікацій. В тому випадку, якщо впевненість класифікації є нижчою за різницю середнього значення та обраної константи, то сегменту буде присвоєний інший клас.

Для того щоб вирішити проблему незбалансованості даних, було використано метод зменшення розмірності. Причому, це могло б допомогти прискорити процес тренування моделей за рахунок зменшення навчальної вибірки. Головна задача цього методу в використанні центроїд які сформовані методом к-середніх на основі якого буде формуватися нова навчальна вибірка меншої розмірності. Але нажаль використання цього методу не дало очікуваного покращеного результату, основною причиною чого є неоднорідність даних.

Також була спроба використання архітектури з більшою кількістю шарів та комірок. В результаті проведеного аналізу стало зрозуміло що комплексність моделей мереж прямого поширення і рекурентної не так сильно залежить від результативної точності класифікації.

Таким чином були протестовані усі методи класифікації, що дозволяє перейти до визначення точності за допомогою метрик які зазначені в формулах 1.4 та 1.5. Для початку потрібно привести точність класифікації не враховуючи інший клас (Таблиця 2.1).

Таблиця 2.1

Точність класифікації без іншого класу

	Метод опорних векторів	Модель випадкового лісу	Нейронна мережа прямого поширення	Рекурентна нейронна мережа
Точність	83.8	92.1	79.	91.2
F1	77.2	86.3	74.2	85.8

За результатами можна бачити, що найбільше точність у методу випадкового лісу. Далі була розрахована точність класифікації з іншим класом (Таблиця 2.2).

Потрібно відмітити, що при включенні іншого класу до навчальної вибірки декотрі алгоритми втрачають можливість класифікувати інші. Вони менш репрезентативні в складі даних тренування класи що і є причиною такого падіння точності класифікації. Говорячи про модель випадкового лісу, можна побачити що вона має достатню точність навіть якщо здійснюється класифікація іншого класу. В той же час, найбільше часу на навчання іде в методу опорних векторів. А метод випадкового лісу, за допомогою паралельності побудови дерев, навчається найшвидше.

Таблиця 2.2

Точність класифікації з іншим класом

	Метод опорних векторів	Модель випадкового лісу	Нейронна мережа прямого поширення	Рекурентна нейронна мережа
Точність	58	87.7	52.2	67
F1	55.2	78.8	498	53

Тому найбільш кращим варіантом є використання моделі випадкового лісу. Досягнуті результати значно покращені в порівнянні з проаналізованими дослідженнями, тому їх можна використовувати.

Висновки до розділу 2

Для того щоб проведення аналізу потоку сенсорів було ефективним, було вибрано та проаналізовано декілька моделей машинного навчання. Як результат аналізу, було виявлено що кожна з моделей може бути використана для використання. Також було показано використання ідеї моделювання тем для цієї проблеми, Таким чином, в кожний сегмент додано інформації щодо імовірнісного розподілу активностей на базі ознак в сегменті. Це було зроблено за допомогою методу Латентного Розміщення Діріхле. За допомогою цього, в сегменті з деякою обмеженістю інформації, буде включено більше інформації про активності. Були показані особливості формування векторів ознак моделей та тренування враховуючи характеристики набору даних. Було визначено, що для даної задачі є деякі характеристики, з якими не є можливим застосувати існуючі підходи. Було наведено результат класифікацій обраних моделей, і за його результатом, якщо ми не використовуємо інший клас, то точність класифікаторів є достатньо високою, та якщо включено інший клас, то точність падає. Основною причиною цього є те що класифікатори втрачають можливість класифікувати деякі класи. Найкращим варіантом для використання є модель випадкового лісу, точність цієї моделі значно краще чим в аналогу, що дає перспективу для використання цього методу для вирішення поставленої задачі.

3. РОЗРОБКА СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ РОЗУМНОГО БУДИНКУ

Розробка програмної системи починається з вибору засобів розробки, оскільки вони впливають на якість кінцевого продукту. Тому, швидкість роботи системи може залежати від того яку бібліотеку машинного навчання було вибрано, оскільки існуючі бібліотеки мають різні реалізації алгоритмів.

Також дуже важливим фактором є вибір мови програмування, адже в кожній мові є свої плюси та мінуси. Наприклад мова Python добре пристосована для задач машинного навчання, крім цього, досить багато бібліотек для роботи з нейронними мережами доступні саме в Python.

Крім того, важливо реалізувати мінімалістичний інтерфейс користувача для створення комфортних умов використання системи. Важливо також надати загальну схему роботи системи, діаграми класів та структуру системи та для пояснення особливостей архітектури програмного забезпечення розробленої системи.

3.1 Програмні засоби розробки

Розробка даного рішення велася у середовищі розробки PyCharm, використовуючи Python 3.6. Python – об'єктно-орієнтована, інтерпретована мова програмування з динамічною типізацією. Ця мова використовується в багатьох напрямках, таких як веб розробка, наукові підрахунки, аналітика та машинне навчання. Основною причиною вибору саме цієї мови програмування є те що доступна велика кількість бібліотек для роботи з машинним навчанням.

Для обробки даних було використано бібліотеку Pandas, що є бібліотекою з відкритим кодом. Ця бібліотека надає ряд можливостей щодо більш зручної роботи з даними. Основною структурою даних є DataFrame, дуже важливим

фактором є те що вона підтримує індексацію. На рисунку 3.1 зображено приклад даної структури.

The diagram illustrates a data frame structure with the following components:

- index labels:** A vertical list of integers from 0 to 9 on the left side of the table.
- column names:** A horizontal list of column headers at the top: Mountain, Height (m), Range, Coordinates, Parent mountain, First ascent, Ascents bef. 2004, and Failed attempts bef. 2004.
- data:** The main body of the table containing rows of data for various mountains.

	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59'17"N 86°55'31"E	NaN	1953	>>145	121,0
1	K2 / Qogir / Godwin Austen	8611	Elahoro Karakoram	35°02'33"N 76°30'45"E	Mount Everest	1954	45	44,0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42'12"N 88°08'51"E	Mount Everest	1955	38	24,0
3	Lhotse	8516	Mahalangur Himalaya	27°57'42"N 86°55'59"E	Mount Everest	1955	26	25,0
4	Makalu	8485	Mahalangur Himalaya	27°53'23"N 87°05'20"E	Mount Everest	1955	45	52,0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05'39"N 86°39'39"E	Mount Everest	1954	70	28,0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41'48"N 83°29'35"E	K2	1960	51	39,0
7	Manaslu	8163	Manaslu Himalaya	28°33'00"N 84°33'35"E	Cho Oyu	1968	49	45,0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14'14"N 74°35'21"E	Dhaulagiri	1963	52	67,0
9	Annapurna I	8091	Annapurna Himalaya	28°35'44"N 83°49'15"E	Cho Oyu	1960	36	47,0

Рис. 3.1 Приклад датафрейму

Окремо потрібно виділити можливість щодо запису та читання даних з файлів, яку надає дана бібліотека. Бібліотека відповідає всім вимогам і є добре оптимізованою, тому її можна використовувати в системі.

Також була використана бібліотека з відкритим кодом Scikit-learn. Ця бібліотека реалізує більшість класичних методів машинного навчання, таких як класифікація, регресія, кластеризація. Серед таких методів є також: випадковий ліс, метод опорних векторів тощо. Крім того дана бібліотека має й інші функції обробки даних, мається на увазі нормалізація і трансформація ознак. Сама бібліотека, більшою мірою написана на мові Python, але інша частина написана на Cython. Cython – це мова програмування, яка використовується для інтеграції з кодом мови C. Ця бібліотека довела свою ефективність і є достатньо популярною. Саме тому для реалізації моделі випадкового лісу і методу опорних векторів була обрана ця бібліотека.

Для зручного створення нейронної мережі була застосована бібліотека Keras. Keras є відкритою бібліотекою для зручної розробки моделей глибокого навчання, сама бібліотека була написана мовою Python. Варто зазначити що в даній бібліотеці було реалізовано найчастіше використовувані блоки нейронних мереж. Серед них різні типи шарів, зокрема такі як згорткові або рекурентні, та функції для активації і оптимізатори. Також доступні інструменти для здійснення обробки та трансформації даних. Крім цього реалізовано ряд інструментів щодо здійснення обробки та трансформації даних. Тому використання даної бібліотеки є зручним. Бібліотека Keras використовувалась для побудови рекурентної нейронної мережі та мережі прямого поширення.

Модель Латентного Розміщення Діріхле була використана за допомогою відкритої бібліотеки `zlda`. Більша частина цієї бібліотеки була написана на мові C, тому є достатньо швидкою.

Користувацький інтерфейс був створений за допомогою бібліотеки Dash. Dash – відкрита бібліотека для створення інтерфейсів, часто використовується для задач аналізу даних та моделювання. Застосунки які використовують Dash – це веб-сервера які працюють на базі бібліотеки Flask і використовують HTTP, та JSON формат для передачі даних.

Для відображення графічного інтерфейсу Dash використовує бібліотеку React.js, що є найбільш популярний на даний час javascript бібліотекою. Dash компоненти зберігають стан React компонента, а саме його властивості та значення. Також Dash має набір інструментів для створення нових React компонентів, на основі яких будуть сгенеровані компоненти Dash. В подальшому, користувачу буде достатньо лише змінити аргументи в компоненті.

Таким чином, використання даної бібліотеки є найкращим варіантом при розробці даної системи.

3.2 Загальна схема функціонування системи

Точкою входу в систему є дані, потрібного формату, зібрані з потоку сенсорів, протягом певного проміжку часу. Ці дані складаються з даних тестування, тобто ті що ми отримуємо у реальному часі та даних тренування, що будуть зібрані під час життя мешканця будинку.

Далі, на рисунку 3.2, ви можете бачити схему роботи системи якщо в неї були завантажені дані тренування.



Рис. 3.2. Схема функціонування системи на даних тренування

На першому етапі буде розрахована матриця взаємної інформації та на основі вікон активностей буде побудована модель латентного розміщення Діріхле. Наступним кроком буде сегментація даних на основі яких надалі відбудеться навчання моделі. Після отримання тренерованої моделі, її можна буде використовувати для того щоб класифікувати дані тестування. Далі, на рисунку 3.3 зображено схему роботи системи в тому випадку якщо на вхід придуть дані тестування.

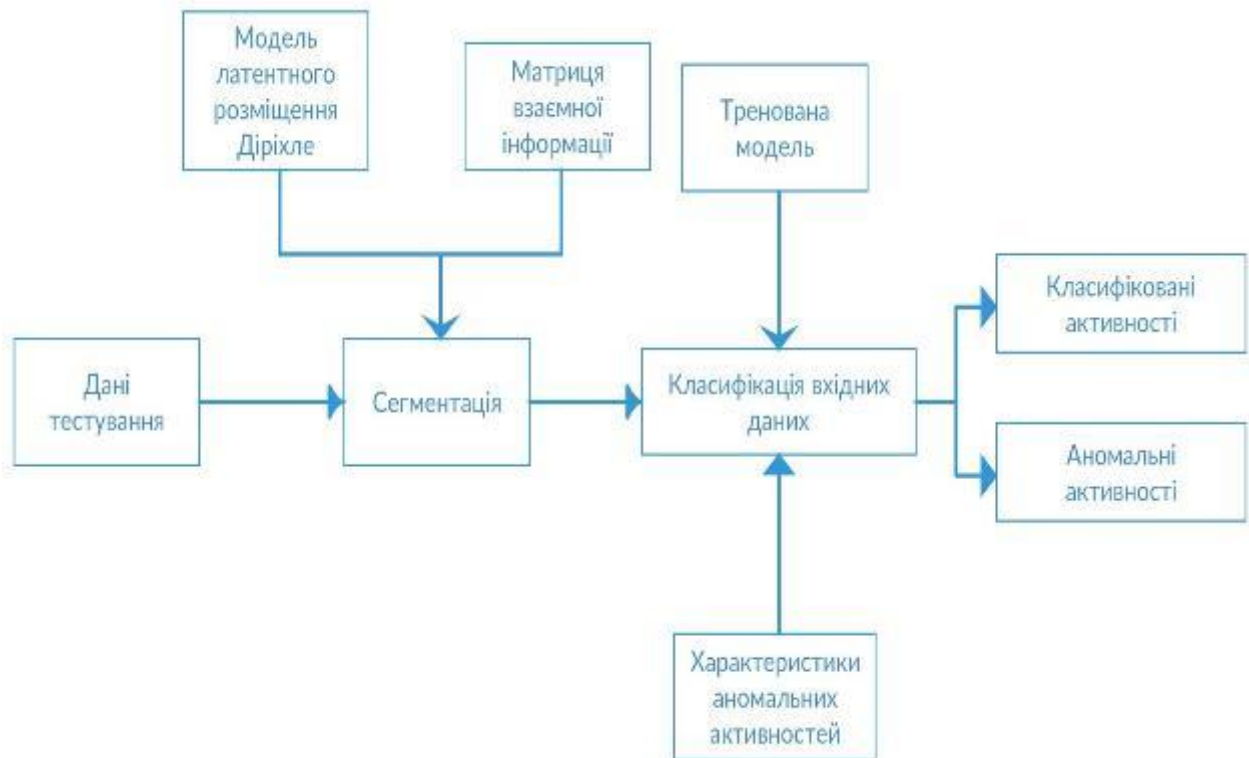


Рис. 3.3. Схема функціонування системи на даних тестування

Як видно зі схеми, якщо на вхід були передані дані тестування то в першу чергу відбудеться сегментація, яка буде використовувати розраховану з даних тренування, матрицю взаємної інформації та побудовану з вікон активностей модель латентного розміщення Діріхле. Наступним кроком відбудеться класифікація сегментів. Після сегментації, дані будуть передані тренерованій моделі. Якщо ж є явно визначені характеристики аномальній активностей, то тоді такі аномальні активності можна буде побачити на виході.

3.3 Сценарії використання системи

Для візуалізації сценаріїв використання системи, було створено діаграму використання. На цій діаграмі явно видно як користувач може взаємодіяти з системою. Діаграму можна побачити на рисунку 3.4.

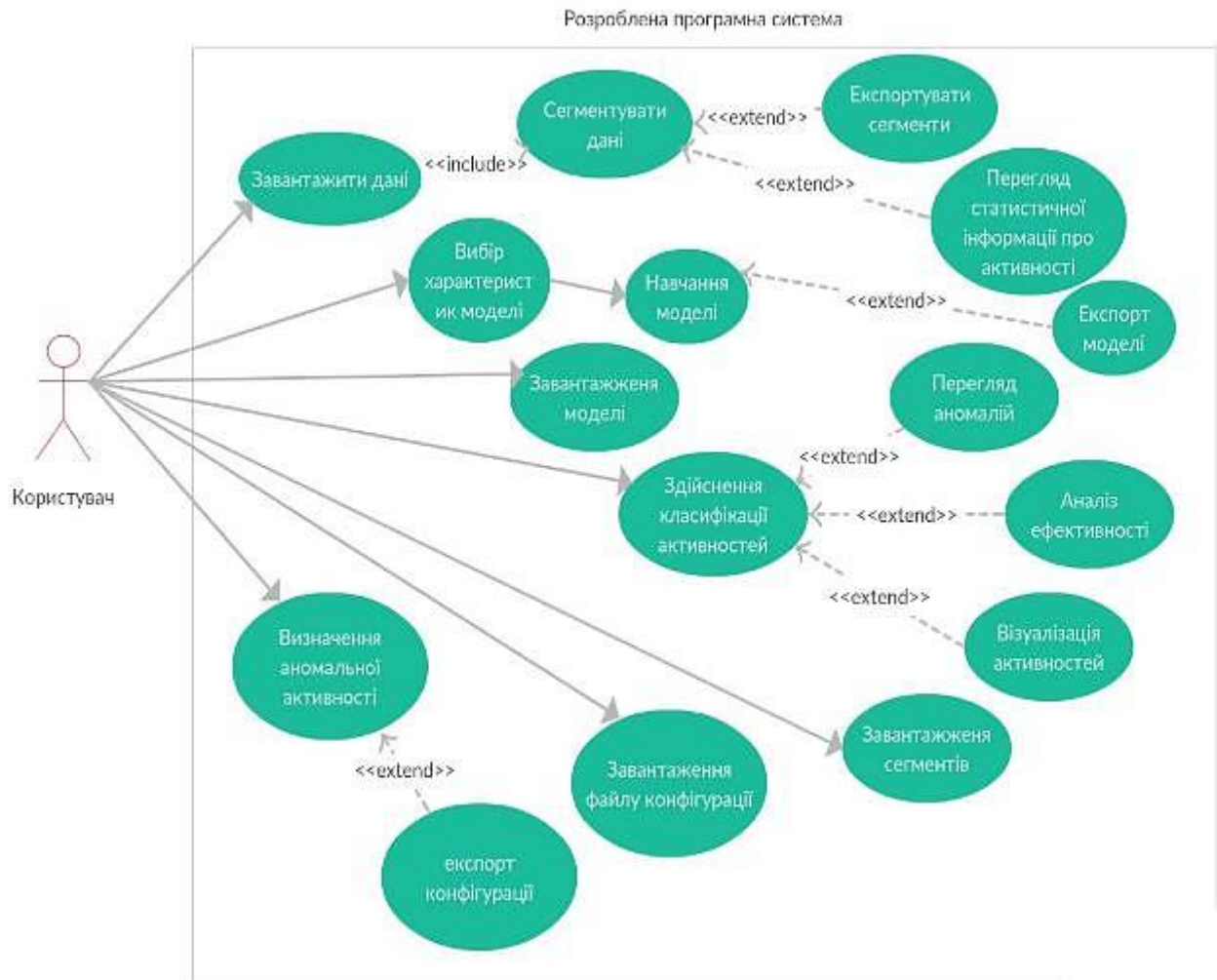


Рис. 3.4. Діаграма використання розробленої системи

Після завершення системою обробки даних тренування і тестування, користувач може завантажити отримані результати, а саме результуючі сегменти даних, розподіл активностей, загальну інформацію та матрицю взаємної інформації.

3.4 Структура програмного забезпечення

На рисунку 3.5 можна побачити основні компоненти програмної системи.

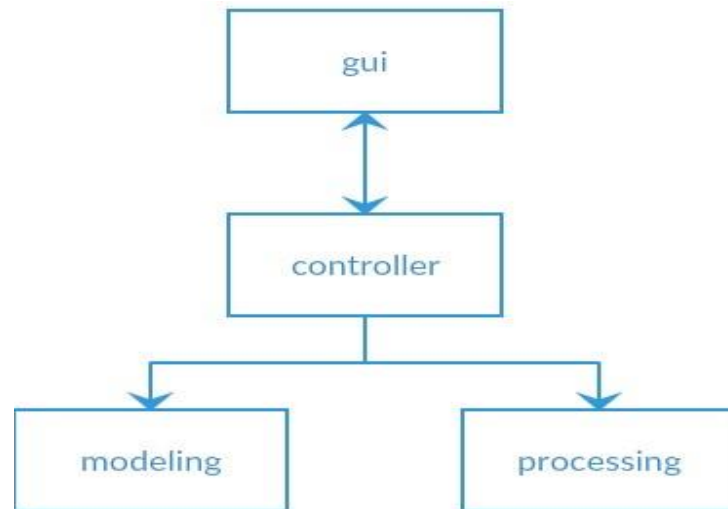


Рис. 3.5. Основні компоненти програмної системи

Загалом програмна система складається з чотирьох компонентів: `gui` відповідає за інтерфейс та візуалізацію, `controller` зв'язує `gui` та `modeling`, `modeling` описує моделі класифікацій, `processing` відповідає за сегментацію.

Також можна побачити діаграму класів на рисунку 3.6. Дана діаграма описує методи та атрибути класів, а також відношення між класами.

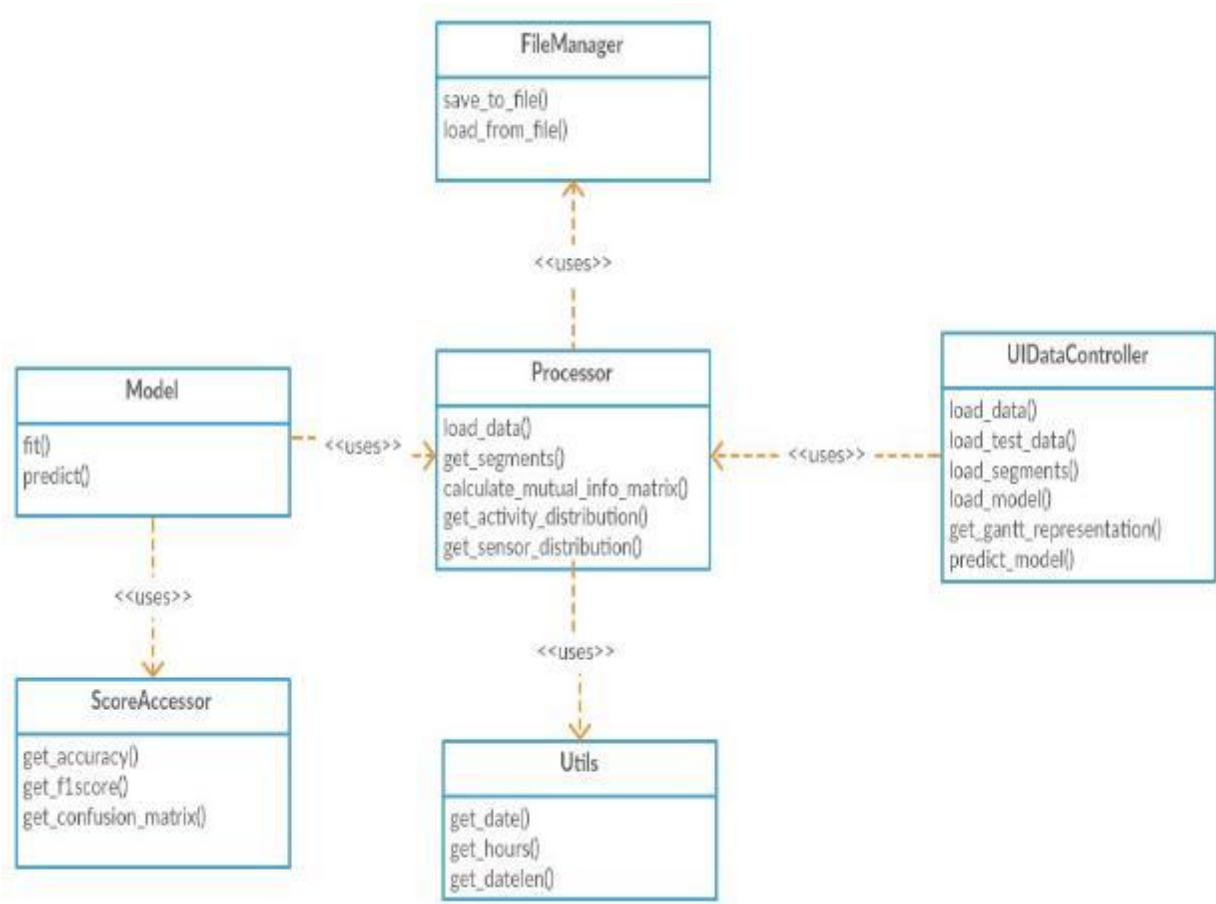


Рис. 3.6. Діаграма класів розробленої системи

Як видно з діаграми, в програмі представлено 5 класів а також основні методи. Клас Processor займається основною роботою щодо обробки даних та розподіляє дані на сегменти. Але в свою чергу клас Processor звертається до класу FileManager для отримання наборів даних. Клас Utils надає деякі методи для обробки даних, здебільшого для роботи з датами. В свою чергу, клас Model містить прогнозуючу модель та основні методи для навчання і прогнозування даних. Клас UIDataController, був розроблений для зв'язку з графічним інтерфейсом і містить загалом методи обробки подій.

Висновки до розділу 3.

В цьому розділі були описані засоби розробки програмної систем, були проаналізовані їх переваги. Було показано чому саме ці засоби розробки були вибрані для розробки даної системи.

Були продемонстровані схеми виконання програми при завантаженні даних тренування та даних тестування. Зокрема при передачі даних тренування, розраховується модель та відбувається тренування моделі класифікацій. А при передачі даних тестування дані будуть сегментовані та класифіковані завдяки інформації отриманій при тренуванні. На виході, користувач отримує перелік класифікацій активностей та може отримати перелік аномальних активностей якщо задані їх характеристики.

Було описано варіанти співпраці користувача з системою за допомогою діаграми використання.

Також були описані всі модулі програмної системи. Було показано взаємозв'язок між класами, та їх основні методи.

4. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З РОЗРОБЛЕНИМ ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

Для зручного використання розробленої системи, було описано технічні вимоги та продемонстровано графічний інтерфейс.

4.1 Технічні вимоги

Для запуску даного застосунку підходить люба операційна система на якій встановлений інтерпретатор Python. До таких операційних систем відноситься Windows, Linux чи macOS. Для того коректного запуску застосунку, мають бути встановлені всі бібліотеки які були описані в розділі 3. Також бажано мати 8 гігабайт оперативної пам'яті, тому що набори даних можуть бути занадто великими.

4.2 Інтерфейс програмного продукту

Розроблена програмна система становить багатосторінковий веб-застосунок, навігація по якому відбувається в меню зверху програми. Меню зображене на рисунку 4.1.

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Confusion](#)

Рис. 4.1 Меню розробленого програмного застосунку

Перше що побачить користувач після запуску програми це Control Panel, на цій панелі користувач має можливість завантажити дані потоку сенсорів, завантажити сегменти або експорту сегментів. Цю сторінку можна побачити на рисунку 4.2.

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Confusion M](#)

Load sensor data

UPLOAD TRAIN DATA

UPLOAD TEST DATA

Load segments data

UPLOAD SEGMENTS X

UPLOAD SEGMENTS Y

UPLOAD DATETIME

Load segments data

UPLOAD SEGMENTS X TEST

UPLOAD SEGMENTS Y TEST

UPLOAD DATETIME

UPLOAD ENCODER

Export data

EXPORT TRAIN SEGMENTS TO DESKTOP

EXPORT TEST SEGMENTS TO DESKTOP

Рис. 4.2 Контрольна панель програмного застосунку

Після здійснення завантаження вхідних даних користувач має можливість переглянути статистичну інформацію про розподіл сенсорів, активностей та загальну інформацію про вхідні дані.

На рисунку 4.3 можна побачити основну інформацію про набір даних. Зокрема кількість днів та кількість активацій сенсорів.

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Confusion Matrix](#)



Рис. 4.3 Загальна статистична інформація

Розподіл сенсорів на даних тестування можна побачити на рисунку 4.4.

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Confusion Matrix](#)

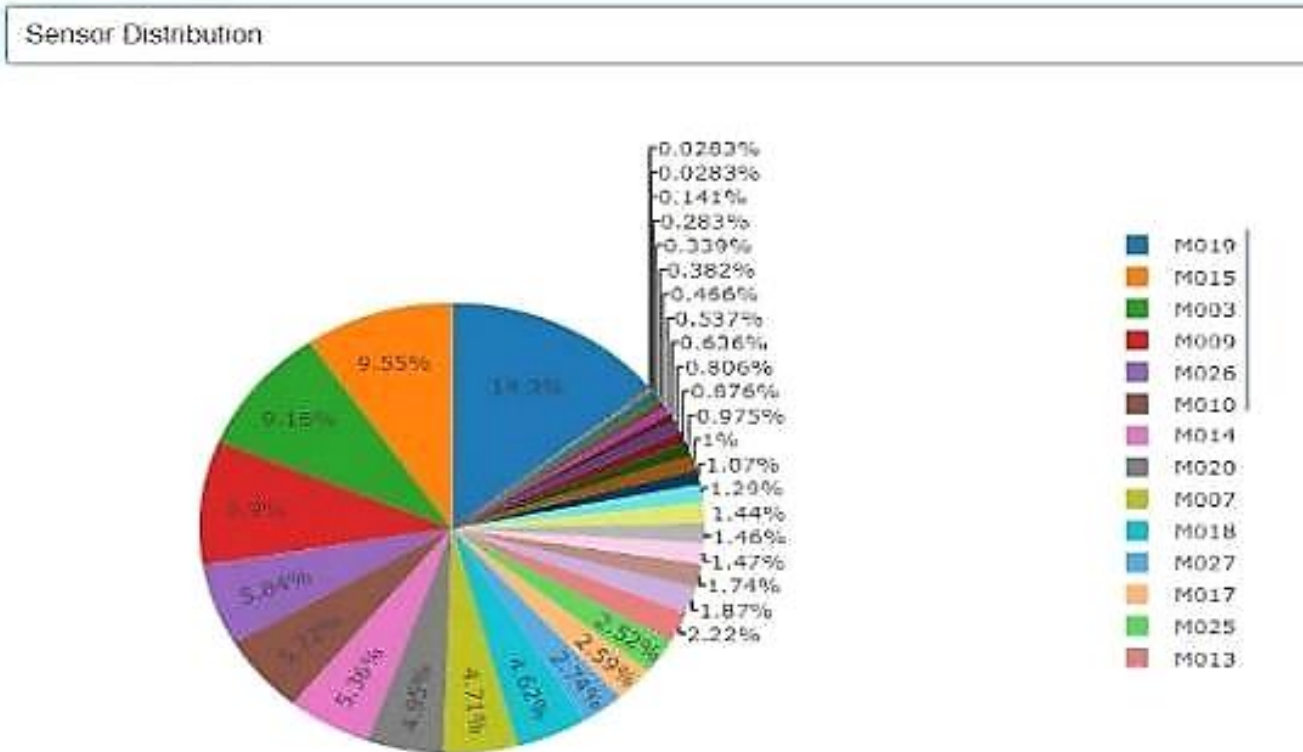


Рис. 4.4 Розподіл сенсорів на даних тренування

Розподіл активностей на даних тестування можна побачити на рисунку 4.5.

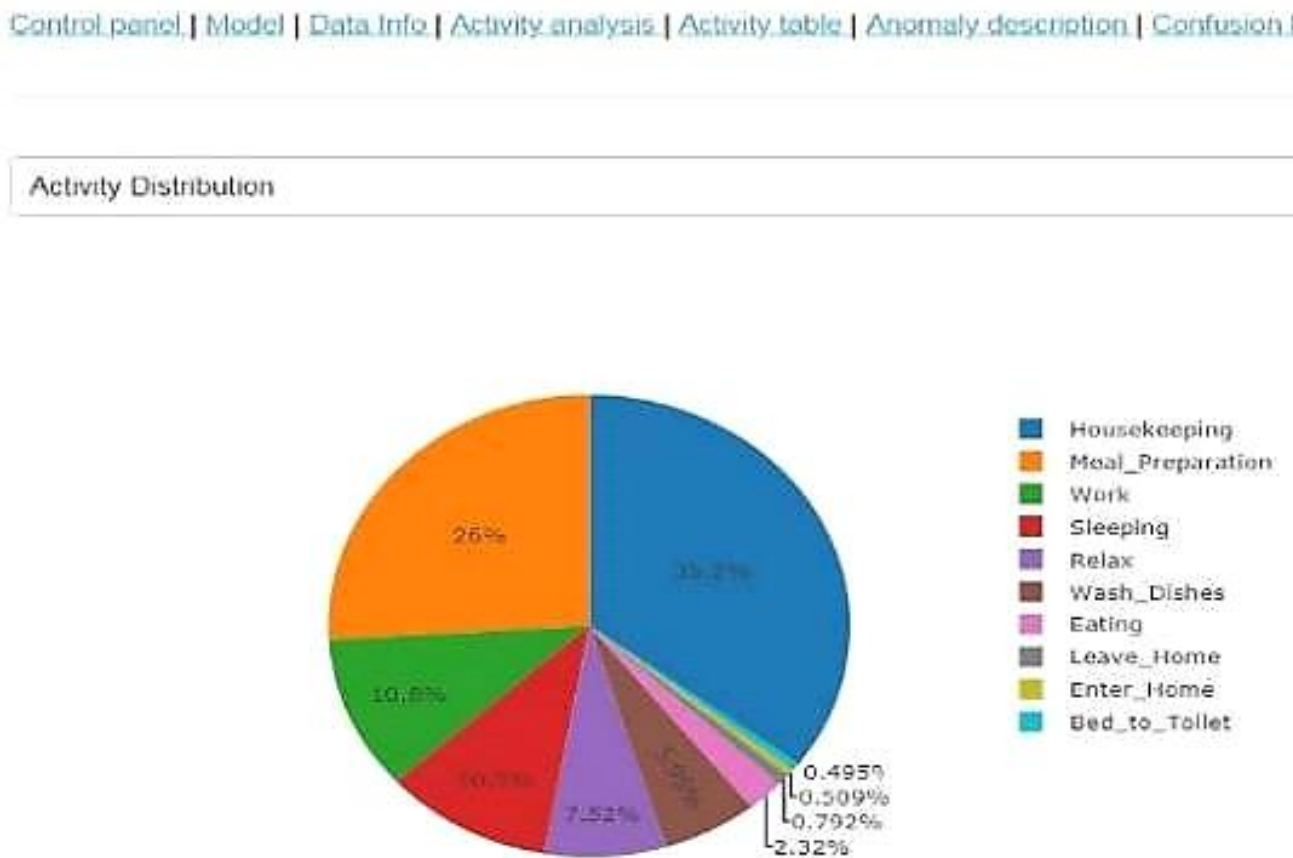


Рис. 4.5 Розподіл активностей для даних тренування

Також у користувача є можливість переглянути матрицю взаємної інформації (Рис. 4.6).

Control panel | Data Info | Activity analysis | **Activity table** | Anomaly description | Confusion Matrix

Mutual Matrix Info																	
	D001	D002	D004	M001	M002	M003	M004	M005	M006	M007	M008	M009	M010	M011	M012	M013	M014
D001	0.01	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
D002	0.0	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D004	0.0	0.0	0.23	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01	0.0	0.0	0.0	0.01	0.01	0.02
M001	0.0	0.0	0.0	0.02	0.0	0.01	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.0	0.02	0.02	0.02
M002	0.0	0.0	0.0	0.0	0.08	0.08	0.0	0.02	0.0	0.08	0.0	0.0	0.0	0.0	0.01	0.0	0.0
M003	0.0	0.0	0.0	0.01	0.08	0.11	0.01	0.03	0.02	0.11	0.02	0.01	0.01	0.0	0.02	0.01	0.02
M004	0.01	0.0	0.01	0.02	0.0	0.01	0.08	0.08	0.03	0.08	0.03	0.02	0.02	0.01	0.02	0.03	0.03
M005	0.01	0.0	0.01	0.02	0.02	0.03	0.08	0.1	0.03	0.1	0.03	0.02	0.03	0.01	0.03	0.03	0.03
M006	0.01	0.0	0.01	0.02	0.0	0.02	0.03	0.03	0.05	0.04	0.05	0.02	0.03	0.01	0.03	0.04	0.04
M007	0.01	0.0	0.01	0.02	0.08	0.11	0.08	0.1	0.04	0.18	0.04	0.02	0.03	0.01	0.03	0.03	0.04
M008	0.01	0.0	0.01	0.02	0.0	0.02	0.03	0.03	0.05	0.04	0.05	0.02	0.03	0.01	0.03	0.04	0.05
M009	0.01	0.0	0.0	0.01	0.0	0.01	0.02	0.02	0.02	0.02	0.02	0.2	0.1	0.01	0.06	0.10	0.05
M010	0.01	0.0	0.0	0.01	0.0	0.01	0.02	0.03	0.03	0.03	0.03	0.1	0.11	0.01	0.06	0.1	0.03
M011	0.01	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
M012	0.01	0.0	0.01	0.02	0.01	0.02	0.02	0.03	0.03	0.03	0.03	0.06	0.06	0.01	0.08	0.08	0.04
M013	0.01	0.0	0.01	0.02	0.0	0.01	0.03	0.03	0.04	0.03	0.04	0.10	0.1	0.01	0.08	0.24	0.11
M014	0.01	0.01	0.02	0.02	0.0	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.03	0.01	0.04	0.11	0.37
M015	0.01	0.0	0.02	0.01	0.0	0.01	0.02	0.02	0.02	0.02	0.03	0.02	0.02	0.01	0.02	0.05	0.19

Рис. 4.6 Перегляд матриці взаємної інформації

Якщо дані тестування вже були завантажені та класифіковані, то у користувача є можливість подивитись істинні активності та класифіковані. Для того щоб переглянути графічне відображення активностей можна перейти на вкладку Activity Analysis. Як будуть відображені істинні активності, можна побачити на рисунку 4.7, а класифіковані активності, можна побачити на рисунку 4.8.

Дуже зручним є те що у користувача є можливість реальні активності та класифіковані, та зобразити їх на одному графіку. Таке порівняння можна побачити на рисунку 4.9.

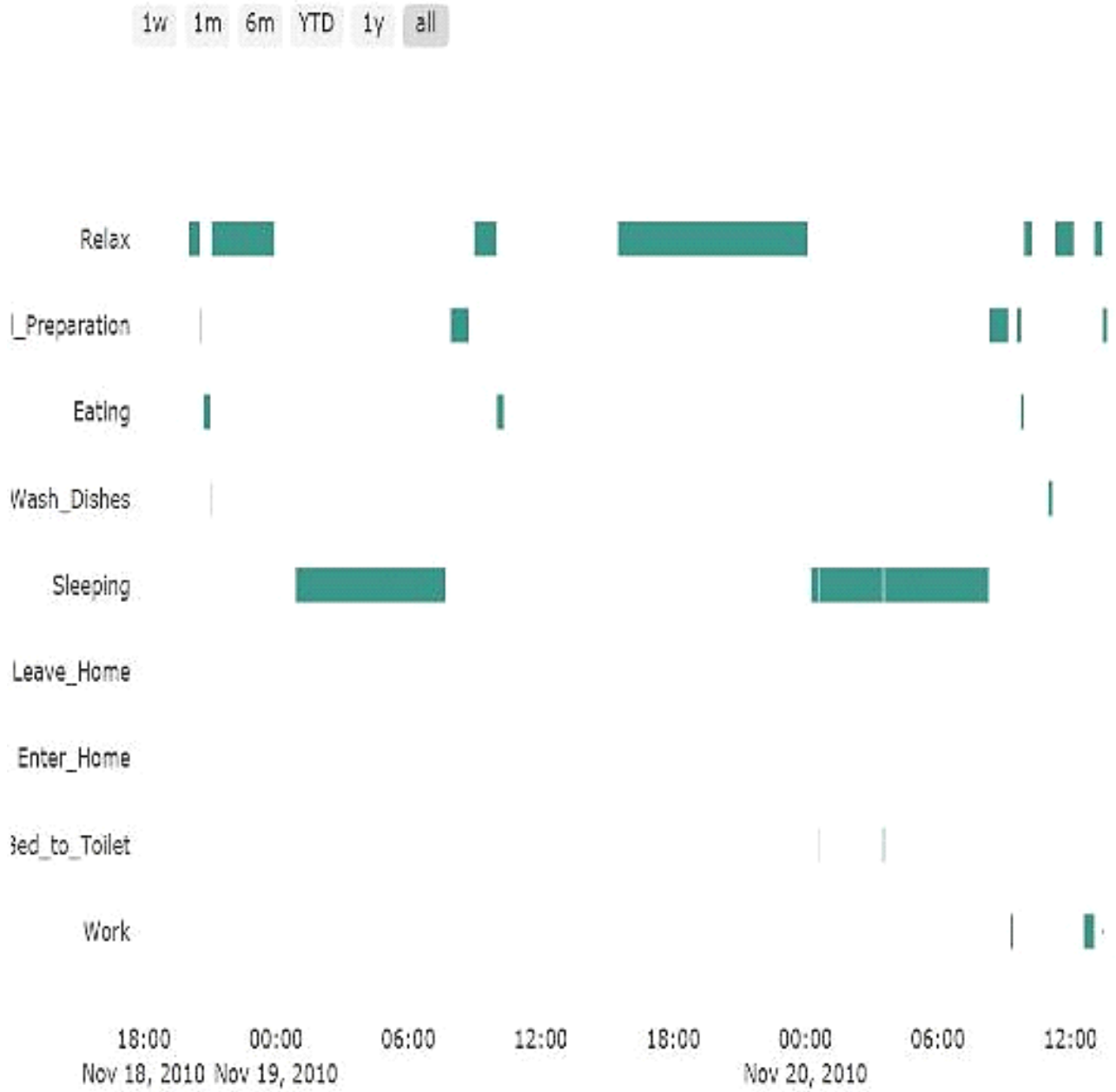


Рис. 4.7 Візуалізація істинних активностей

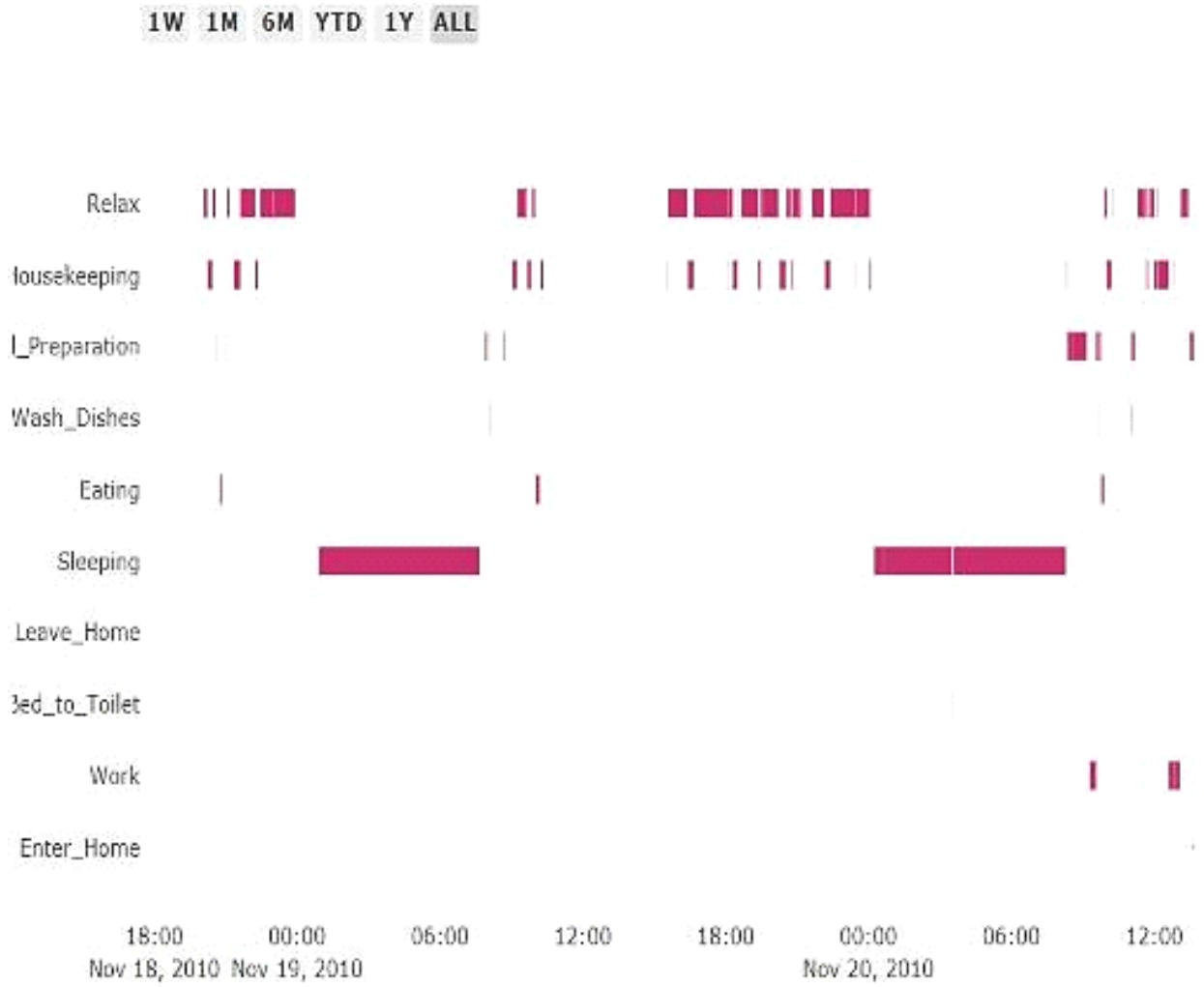


Рис. 4.8 Візуалізація класифікованих активностей

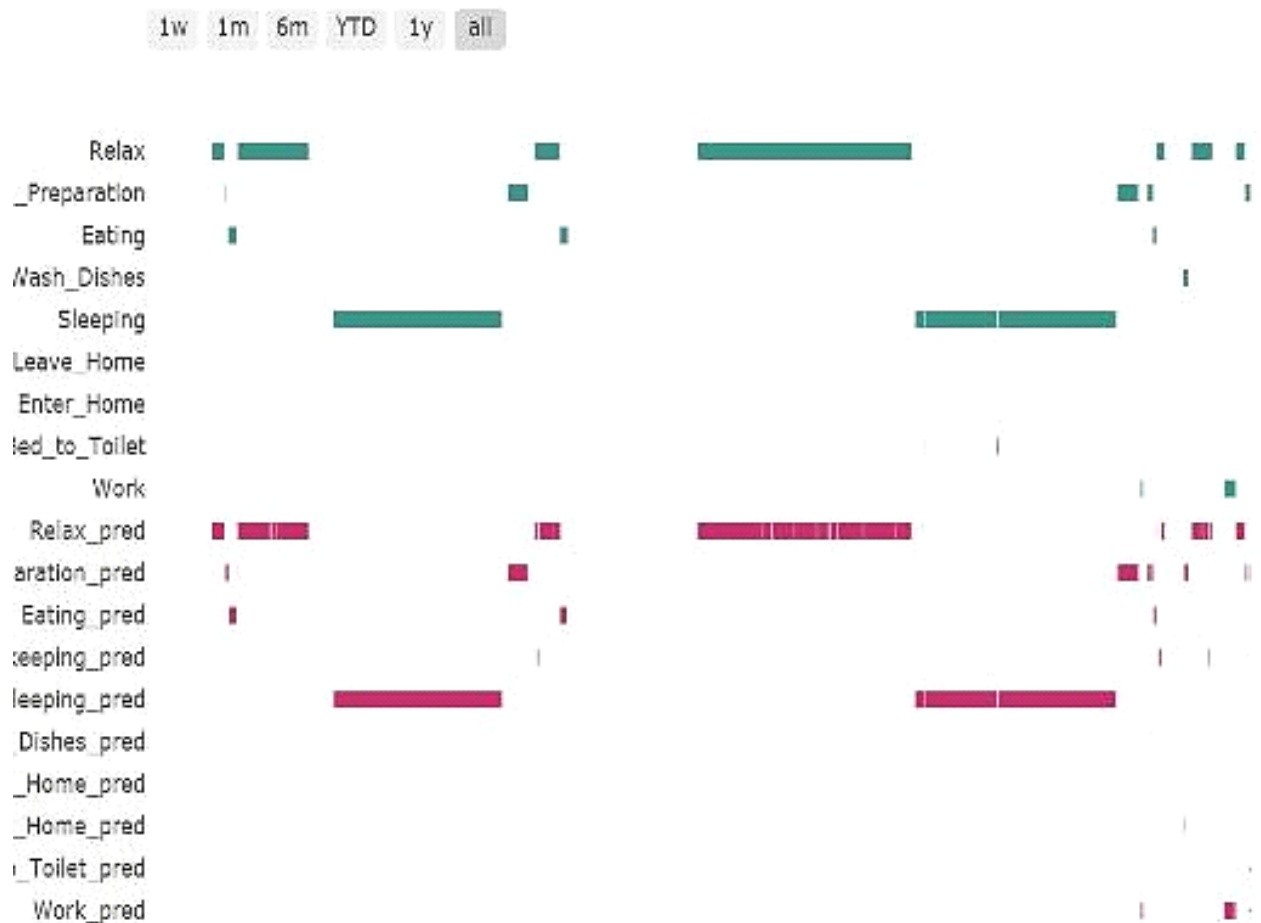


Рис. 4.9 Візуалізація порівняння активностей

Також у користувача є можливість подивитись класифіковані активності у вигляді таблиці. Це можна зробити на вкладці *Activity table*. У цієї таблиці колонками виступають назви активностей, їх початку та кінця. Цю таблицю можна побачити на рисунку 4.10. Також, якщо користувач відкриє вкладку *Confusion Matrix*, то він матиме змогу переглянути матрицю помилок для оцінки ефективності моделі. Приклад матриця помилок можна переглянути на рисунку 4.11.

[Control panel](#) | [Model](#) | [Data info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Comparison Matrix](#)

activity	end	
Relax	2010-11-16 20:04:10	2
Housekeeping	2010-11-16 20:04:40	2
Relax	2010-11-16 20:13:58	2
Housekeeping	2010-11-16 20:23:47	2
Relax	2010-11-16 20:33:49	2
Housekeeping	2010-11-16 20:35:43	2
Meal_Preparation	2010-11-16 20:36:17	2
Meal_Preparation	2010-11-16 20:36:49	2
Wash_Dishes	2010-11-16 20:42:41	2
Housekeeping	2010-11-16 20:47:55	2
Eating	2010-11-16 20:49:59	2
Housekeeping	2010-11-16 21:01:30	2
Meal_Preparation	2010-11-16 21:02:29	2
Meal_Preparation	2010-11-16 21:03:26	2
Relax	2010-11-16 21:08:36	2

Рис. 4.10 Таблиця активностей

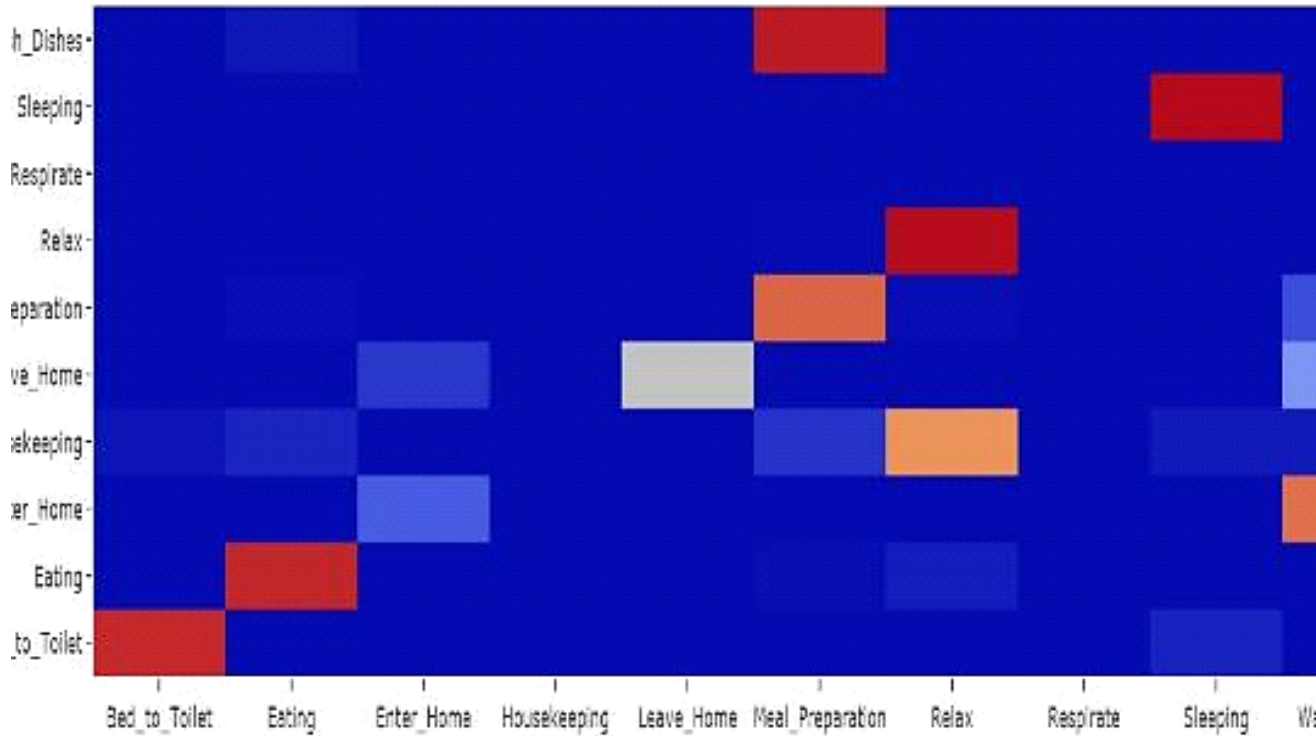


Рис. 4.11. Матриця помилок

Для того щоб визначити характеристики аномальних активностей, користувач може перейти на вкладку Anomaly description.

Якщо ж користувач хоче налаштувати модель класифікацій, то він може зробити це на вкладці Model. Крім того що тут можна вибрати бажану модель з заданими параметрами, тут також можна побудувати її та класифікувати. Експорт моделі також є можливим на цій вкладці.

Вкладку Model можна переглянути на рисунку 4.12.

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#)

The image shows a web interface for configuring a model. At the top, there is a navigation bar with links: Control panel | Model | Data Info | Activity analysis | Activity table | Anomaly description. Below the navigation bar, there is a form with the following elements:

- A dropdown menu with "LSTM" selected.
- A label "Epochs" followed by a text input field containing "10".
- A label "Cell number" followed by a text input field containing "64".
- A button labeled "BUILD CLASSIFIER MODEL".
- A button labeled "PREDICT".
- A button labeled "UPLOAD MODEL".
- A button labeled "EXPORT MODEL".

Рис. 4.12. Вибір моделі класифікації

4.3 Робота з аномальними активностями

Кожна людина має особливі риси, поведінку та звички. Тому найкращим варіантом є дати користувачу самому визначити які активності для нього не притаманні, тобто аномальні.

Для того щоб визначити характеристики аномальних активностей, можна задати дані в формі на вкладці Anomaly Description. Наприклад користувач може задати характеристики для аномалій для вибраної активності. (Рис. 4.13).

[Control panel](#) | [Model](#) | [Data Info](#) | [Activity analysis](#) | [Activity table](#) | [Anomaly description](#) | [Confus](#)

Config creation

Disruption value

Activity

Add condition less

Add condition more

Add time condition
From To

Рис. 4.13 Вкладка створення конфігурації аномалій

Таким чином користувач може наприклад вказати тривалість аномальної активності, тобто якщо тривалість виходить за рамки нормального діапазону. Також користувач може вказати часовий інтервал, у межах якого, активність може бути аномальною.

Корисним є те, що користувач може визначити який час має пройти для того щоб активності вважалися окремими. Наприклад, якщо людина проснулася посеред ночі на 10 хвилин,, але потім лягла спати знову, то не має сенсу розділяти такі активності, а визначити їх аномальними.

Якщо користувач налаштував ці конфігурації, то він має можливість експортувати їх собі, тому що вони можуть ще пригодитись. На рисунку 4.14 показано експортований файл конфігурації.

```
{
  "time_disrupt_mins": "45",
  "len_less": [
    {
      "activity": "Sleeping",
      "value": "8"
    }
  ],
  "len_more": [
    {
      "activity": "Sleeping",
      "value": "12"
    }
  ],
  "time_interval": [
    {
      "activity": "Sleeping",
      "lower_bound": "15",
      "upper_bound": "17"
    }
  ]
}
```

Рис. 4.14 Файл конфігурації аномальних активностей

Якщо користувач описав аномальні активності то у нього буде можливість переглянути всі аномальні активності серед класифікованих. Список аномальних активностей можна побачити на рисунку 4.15.

activity	length	start
Sleeping	6.0	2011

Рис. 4.15. Перелік аномалій активностей

Висновки до розділу 4

В цьому розділі було розглянуто методику роботи розробленої програми. Розглянуто технічні вимоги для використання розробленої програмної системи. Було продемонстровано всі вкладки графічного інтерфейсу розробленої програми з їх детальним описом.

Описано як користувач може задавати характеристики аномальних активностей.

В результаті, дана програмна система дає користувачу багато різних функцій та проводити аналітику вхідних даних. Зокрема вибирати модель класифікацій зі вказаними характеристиками. Тож користувач зможе проаналізувати аномальні активності та оцінити отримані результати.

5. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ВПРОВАДЖЕННЯ ПРОЕКТУ

Для підтвердження того що даний проект є корисним буде проаналізована можливість розробки стартап проекту для даної системи аналізу даних. Так як програмна система направлена на збір даних щодо життя людини, то це може бути дуже корисним то буде проаналізовано чи можна розробити стартап на базі даної системи. Потрібно буде здійснити аналіз та визначити особливості виходу даної ідеї на ринок. Таким чином, розроблену програмну систему будуть використовувати розробники шлюзів розумного будинку. Шлюз являє собою головний пристрій в розумному будинку. Він контролює обмін даними між пристроями, і саме ці дані і будуть точкою входу в систему.

5.1 Опис ідеї проекту

Потрібно описати Важливо описати, яку користь розроблювана система може принести різним категоріям користувачів. В таблиці 5.1 описані ідеї проекту.

Таблиця 5.1

Опис ідеї проекту

Зміст ідеї	Напрямки застосування	Сегменти споживачів	Цінність для споживачів
Аналіз даних в рамкаї розумного будинку.	1. Конфігурація розумного будинку для потреб мешканців.	Масовий ринок (Мешканці розумних будинків).	Покращення комфорту проживання в розумному будинку.
	2. Застосування у соціальній сфері та сфері охорони здоров'я.	Нішевий (Соціальні працівники, медпрацівники).	Можливості моніторингу за людьми з вадами.

5.2 Технологічний аудит проекту

Далі потрібно проаналізувати технологічну реалізованість проекту. Для цього треба описати ідеї та їх можливості реалізації. Технологічний аудит можна побачити в таблиці 5.2.

Таблиця 5.2

Технологічний аудит проекту

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Система аналізу даних жителів будинку	Набір алгоритмів машинного навчання	Необхідність покращити алгоритми.	Технологія ще має бути розробленою.
2	Розробка API та реалізація програм	Мови програмування.	Технологія існує.	Технологія доступна.

Як ми бачимо із таблиці для технологічної реалізації проекту, потрібно розробити API для клієнтських програм. Це можливо зробити, але спочатку потрібно буде створити систему аналізу на основі існуючих алгоритмів.

Також важливо проаналізувати можливі загрози проекту, щоб правильно вибрати напрямок розвитку.

Спочатку треба перевірити попит на проект, його наявність, динаміку та обсяг. Характеристику ринку можна побачити у таблиці 5.3.

Таблиця 5.3

Попередня характеристика стартап проекту

Показники стану ринку	Характеристика
Потреба в продукції	Існує
Можливі річні обсяги випуску в натуральних показниках	Необмежено
Річні обсяги випуску в вартісних показниках	20000–30000\$
Динаміка ринку	Зростає
Наявність обмежень для входу	Проблема конфіденційності даних
Середня норма рентабельності в галузі (або по ринку), %	72%

Виходячи з інформації представленої в даній таблиці, даний ринок є досить привабливим. Однак головною проблемою є конфіденційність, адже система напряду працює з даними мешканців будинку.

Одним з найважливіших етапів є аналіз та виділення клієнтів та їх потреб. Цю характеристика представлено у таблиці 5.4.

Таблиця 5.4

Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Особливості поведінки споживачів	Вимоги споживачів до товару
Інтелектуальний аналіз даних.	Жителі розумних будинків.	Користувачі не мають сервісів для налаштувань та конфігурацій будинків.	Ефективність функціонування системи та ціна.
	Люди з вадами здоров'я та похилого віку	Є потреба в моніторингу за мешканців з можливістю.	Отримання списку аномальних активностей.

Далі потрібно проаналізувати конкурентоспроможність проекту для можливості виходу на ринок. Таким чином, необхідно оцінити прямих та потенційних конкурентів. Але слід зазначити що прямих конкурентів не має.

Також необхідно провести SWOT аналіз проекту. (таблиця 5.5).

Таблиця 5.5

SWOT аналіз проекту

— — —	Сильні сторони (S): Розробники шлюзів надають доступ до клієнтської бази Постійний ріст якості сервісу з ростом кількості клієнтів	— — —	Слабкі сторони (W): Необхідно тримати в секреті результати досліджень Потрібно зібрати певний масив даних від мешканців перед тим як сервіс буде функціонувати
— — —	Можливості (O): Заклучення більш вигідних контрактів при розвитку проекту. Постійний ріст ринку розумних домів. Відсутність конкурентів.	— — —	Загрози (T): Конфіденційність даних мешканців. Можлива поява конкурентів. Занепад ринку розумних будинків.

На основі результатів проведеного SWOT-аналізу потрібно розробити альтернативу ринкової поведінки для запуску стартап-проекту на ринок. Також необхідно визначити час їх реалізації на ринку.

Також потрібно визначити альтернативи ринкового впровадження стартап-проекту, такі приведені у таблиці 5.6. Таким чином, інвестиції можуть бути отримані з двох джерел: інвесторів та через збирання кошти на краудфандингових платформах.

Таблиця 5.6

Альтернативи ринкового впровадження стартап-проекту

Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Пошук інвесторів	78%	4–6 місяців
Використання платформ краудфандінгу	66%	Більше року

В даному випадку більш доцільним буде пошук інвесторів, тому що це займе менше часу та можливої кількості наданих ресурсів.

5.3 Розроблення ринкової стратегії проекту

Для розроблення ринкової стратегії спочатку потрібно описати цільові групи потенційних споживачів. Цільові групи описані в таблиці 5.7.

Таблиця 5.7

Вибір цільових груп потенційних споживачів

Опис цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в сегменті	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Жителі розумного будинку	Потребують	Попит наявний	Конкуренти відсутні	Просто
Люди з вадами здоров'я, люди похилого віку	Потребують	Попит наявний	Конкуренти відсутні	Помірно

Якщо враховувати попит та відсутність конкурентів, то зрозуміло що потрібно працювати з всіма групами.

Сформуємо стратегію розвитку, яка представлена у таблиці 5.8.

Таблиця 5.8

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Пошук інвесторів	Стратегія диференціації	Охоплення декілька різних груп користувачів.	Стратегія диференціації

Також треба сформуванати маркетингову концепцію товару. Ключові переваги концепції потенційного товару приведено у таблиці 5.9.

Таблиця 5.9

Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
Персоналізація будинку	Покращення умов проживання будинку.	Конкуренти відсутні.
Моніторинг життя мешканців	Можливість моніторингу активності мешканців та виявлення аномальних активностей.	Конкуренти відсутні.

Враховавши доходи потенційних власників розумних будинків можна визначити верхню та нижню межі встановлення ціни (таблиця 5.10).

Таблиця 5.10

Визначення меж встановлення ціни

Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни
–	1500–6000 \$	40–95 \$

Важливим кроком є визначення системи збуту (таблиця 5.11).

Таблиця 5.11

Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти мають мати можливість	Створення мосту між даними з	Одно рівневий – тільки через	В якості місячної підписки на

випробовування системи перед придбанням.	шлюзу будинку і розробленою системою	розробників шлюзів	сервіс.
--	--------------------------------------	--------------------	---------

Отримані результати говорять про те що сервіс буде розповсюджуватись як платна підписка, з можливістю безкоштовного тестування. Ціна встановлена з огляду на середній дохід потенційних користувачів.

Висновок до розділу 5

У рамках цього розділу було проаналізовано особливості створення стартап проекту для розробленої системи. Великим плюсом є те що на ринку не існує сервісів які б надавали подібні можливості.

Таким чином, при постійному рості ринку та відсутність конкурентів, можна дійти до висновку, що вихід на ринок цілком можливий. Для цього на початку стартапу потрібно довести клієнтам що технологія буде їм корисною. Цей проект має дві потенційні групи клієнтів: людей з вадами здоров'я, мешканців розумних будинків. Технічні особливості цього проекту надають можливість роботи обома групами покупців.

Було проведено аналіз можливих альтернатив виходу на ринок, за результатами якого було вибрано вийти через інвестиції. Тому що їх можна швидше отримати порівняно з іншим варіантом. На основі наданого аналізу можна зробити висновок, що подальша імплементація проекту є можливою.

ВИСНОВКИ

В результаті виконання магістерської роботи можна зробити наступні висновки:

Проаналізовані існуючі в дослідженнях методи та підходи щодо здійснення аналізу активності людини в розумному будинку у реальному часі використовуючи дані бінарних сенсорів руху. Запропоновані в дослідженнях методи стали базою для подальшого удосконалення. Описані особливості наборів даних, на які необхідно звернути увагу при удосконаленні методу.

Удосконалено метод розпізнавання активності на базі даних бінарних сенсорів руху у реальному часі за рахунок використання моделі випадкового лісу в якості класифікатора і моделі Латентного Розміщення Діріхле для включення в сегмент інформацію про розподіл активностей, що призвело до зростання точності класифікації. У середньому можна відмітити ріст точності класифікації на 20%. Також описані та проаналізовані результати експериментів з рядом інших підходів, а також точність класифікації інших моделей.

Розроблена програмна система, яка використовуючи удосконалену модель може здійснювати візуалізацію результатів розпізнавання і аналізу активностей на аномалії. Для розробки програмної системи були обрані засоби та бібліотеки з врахуванням результативної ефективності. Описана програмна структура розробленої системи використовуючи діаграми пакетів та класів, сценарії використання системи на базі діаграми активностей, а також наведено методику роботи користувача з системою.

Враховуючи можливість функціонування у реальному часі – розроблена система може бути використана для здійснення автоматизації розумного будинку. Іншим важливим застосунком системи є моніторинг за життям людей з вадами здоров'я. Для такого застосунку у розробленій системі передбачена можливість визначення аномальних активностей.

Розроблений стартап проект який включає в себе багатофакторний аналіз можливості виходу проекту на ринок. У результаті аналізу ціннісної пропозиції, яку несе система, можливості технічного впровадження, потенційних сегментів клієнтів та їх потреб до програмної системи, конкурентного становища на ринку, можливих потенційних конкурентів, сильних та слабких сторін проекту, можливих альтернатив ринкового впровадження та особливостей здійснення маркетингової компанії було зроблено висновок що є доцільним подальша імплементація проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Малишев М. С. Інтелектуальний аналіз даних в умовах «розумного будинку» / М. С. Малишев, Ю. А. Тарнавський. Сучасні проблеми наукового забезпечення енергетики / М. С. Малишев, Ю. А. Тарнавський., 2019. С. 195.
2. Розроблення стартап-проекту. Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О. А. Гавриша. Київ: НТУУ «КПІ», 2017. 28 с.
3. Тарнавський Ю. А. Інтелектуальний аналіз даних в умовах розумного будинку / Ю. А. Тарнавський, М. С. Малишев. Сталий розвиток – XXI століття: управління, технології, моделі / Ю. А. Тарнавський, М. С. Малишев., 2019. С. 448–450.
4. A Novel Connectionist System for Unconstrained Handwriting Recognition / [A. Graves, M. Liwicki, S. Fernandez та ін.]. IEEE Transactions on Pattern Analysis and Machine Intelligence / [A. Graves, M. Liwicki, S. Fernandez та ін.]. 2010. С. 855–868.
5. Abdallah Z. Activity Recognition with Evolving Data Streams: A Review / Z. Abdallah, M. Medhat. ACM Computing Surveys (CSUR) / Z. Abdallah, M. Medhat., 2019. С. 1–36.
6. An experimental study on new features for activity of daily living recognition / D. Ferretti, E. Principi, S. Squartini, L. Mandolini. International Joint Conference on Neural Networks (IJCNN) / D. Ferretti, E. Principi, S. Squartini, L. Mandolini., 2017.
7. Beckmann M. A KNN Undersampling Approach for Data Balancing / M. Beckmann, N. Ebecken. *Journal of Intelligent Learning Systems and Applications* / M. Beckmann, N. Ebecken., 2016. С. 104–116.
8. Belgiu M. Random forest in remote sensing: A review of applications and future directions / M. Belgiu, L. Drãgut. *ISPRS Journal of Photogrammetry and Remote Sensing* / M. Belgiu, L. Drãgut., 2016. С. 24–31.

9. Belgiu M. Random forest in remote sensing: A review of applications and future directions / M. Belgiu, L. Dragut. *ISPRS Journal of Photogrammetry and Remote Sensing* / M. Belgiu, L. Dragut., 2017. С. 24–31.
10. Bishop C. *Pattern Recognition and Machine Learning* / Christopher Bishop. New York, 2007. 738 с.
11. Blei D. Latent Dirichlet Allocation / D. Blei, A. Ng, M. Jordan. *Journal of Machine Learning Research* / D. Blei, A. Ng, M. Jordan., 2004. С. 993–1022.
12. Blei D. Probabilistic Topic Models / David Blei. *Communications of the ACM* / David Blei., 2013. С. 77–84.
13. Breinman L. *Classification and regression trees* / L. Breinman, J. Friedman, R. Olshen. Wadsworth & Brooks / Cole Advanced Books & Software / L. Breinman, J. Friedman, R. Olshen. 1984.
14. CASAS: A smart home in a box / D. Cook, A. Crandall, B. Thomas, N. Krishnan. *IEEE Computer* / D. Cook, A. Crandall, B. Thomas, N. Krishnan., 2014. С. 26–33.
15. Chen D. *Pandas for Everyone: Python Data Analysis*. Daniel Chen., 2018. 406 с.
16. Chih-Chung C. *LIBSVM: A Library for Support Vector Machines* / C. Chih-Chung, Chih-Jen., 2002. 39 с.
17. Chih-Wei H. A Comparison of Methods for Multi-class Support Vector Machines / H. Chih-Wei, L. d Chih-Jen. *IEEE Transactions on Neural Networks* / H. ChihWei, L. d Chih-Jen., 2003. С. 415–425.
18. Cook D. Activity recognition on streaming sensor data / D. Cook, N. Krishnan. *Pervasive and Mobile Computing* / D. Cook, N. Krishnan., 2015. С. 138–154.
19. Cook D. Collecting and disseminating smart home sensor data in the CASAS project / D. Cook, M. Schmitter-Edgecombe, C. A. *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research* / D. Cook, M. Schmitter-Edgecombe, C. A., 2010.

20. Discriminant Measures for Performance Evaluation / M. Sokolova, N. Japkowicz, S. Szpakowicz., 2007. С. 1015–1021.
21. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. №15, 2015. С. 1929–1958.
22. Fahad L. Activity Recognition in Smart Homes Using Clustering Based Classification / L. Fahad, S. Tahir, M. Rajarajan. *22nd International Conference on Pattern Recognition* / L. Fahad, S. Tahir, M. Rajarajan., 2015.
23. Fowler M. UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language / M. Fowler, K. Scott., 1999. 224 с.
24. Garreta R. Learning scikit-learn: Machine Learning in Python / R. Garreta, G. Moncecchi., 2014. 118 с.
25. Gemino A. Use case diagrams in support of use case modeling: Deriving understanding from the picture / A. Gemino, D. Parker // *Journal of Database Management* / A. Gemino, D. Parker., 2010. С. 1–24.
26. Goodfellow I. Deep Learning (Adaptive Computation and Machine Learning) / I. Goodfellow, B. Yoshua, A. Courville. 2017. 775 с.
27. Grinberg M. Flask Web Development: Developing Web Applications with Python / Miguel Grinberg. 2015. 258 с.
28. Gulli A. Deep Learning with Keras: Implementing deep learning models and neural networks with the power of Python / Antonio Gulli., 2018. 318 с.
29. Iterative Design of Visual Analytics for a Clinician-in-the-loop Smart Home. / [A. Ghods, K. Caffrey, D. Cook та ін.]. *IEEE Journal of Biomedical and Health Informatics* / [A. Ghods, K. Caffrey, D. Cook та ін.]. 2019.
30. Jedrzejowicz P. Stacked Ensemble Models for Improved Prediction Accuracy / P. Jedrzejowicz, I. Czarnows. *IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)* / P. Jedrzejowicz, I. Czarnowsk. 2018.
31. Jeffreys H. Scientific Inference / Harold Jeffreys., 2012. 282 с.

32. Kasteren T. Hierarchical Activity Recognition Using Automatically Clustered Actions / T. Kasteren, G. Englebienne, B. Krose. International Joint Conference on Ambient Intelligence / T. Kasteren, G. Englebienne, B. Krose., 2012. С. 82–91.
33. Kingma D. Adam: A Method for Stochastic Optimization / D. Kingma, J. Ba. International Conference on Learning Representations / D. Kingma, J. Ba., 2016.
34. Kirienko, A. Human Behaviour Analysis in Context of Smart Environment Automation / A. Kirienko, I. Soloviev, Computer tools in education / A. Kirienko, I. Soloviev, St. Petersburg, 2018. С. 15–29
35. Krishnan N. Activity recognition on streaming sensor data / N. Krishnan, D. Cook. Pervasive and Mobile Computing / N. Krishnan, D. Cook., 2014. С. 138–155.
36. LSTM: A Search Space Odyssey / [K. Greff, K. Srivastava, J. Koutnik та ін.]. Transactions on neural networks and learning systems 1 / Greff, K. Srivastava, J. Koutnik та ін.], 2016. С. 2222–2232.
37. Machalot F. Activity Recognition in Sensor Data Streams for Active and Assisted Living Environments / F. Machalot, A. Haj Mosa, A. Mouhannad. IEEE Transactions on Circuits and Systems for Video Technology / F. Machalot, A. Haj Mosa, A. Mouhannad., 2018. – С. 2933–2945.
38. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython / Wes McKinney., 2018. 550 с.
39. Mehr H. Resident activity recognition in smart homes by using artificial neural networks / H. Mehr, P. Huseyin, A. Cetin. International Istanbul Smart Grid Congress and Fair (ICSG) / H. Mehr, P. Huseyin, A. Cetin., 2017.
40. Nawel Y. Feature extraction for human activity recognition on streaming data / Y. Nawel, F. Belkacem, A. Fleury. International Symposium on Innovations in Intelligent Systems and Applications (INISTA) / Y. Nawel, F. Belkacem, A. Fleury. 2016.

41. Nawel Y. Towards improving feature extraction and classification for activity recognition on streaming data / Y. Nawel, F. Belkacem, A. Fleury. *Journal of Ambient Intelligence and Humanized Computing* / Y. Nawel, F. Belkacem, A. Fleury., 2017. С. 177–189.
42. Novakovic J. Using Information Gain Attribute Evaluation to Classify Sonar Targets / Jasmina Novakovic. *17th Telecommun. Forum TELFOR* / Jasmina Novakovic., 2010. С. 24–26.
43. Park J. *IEEE 4th World Forum on Internet of Things (WF-IoT)* / J. Park, J. Kiyong, Y. Sung-Bong. Deep neural networks for activity recognition with multi-sensor data in a smart home / J. Park, J. Kiyong, Y. Sung-Bong., 2019.
44. Patle A. SVM kernel functions for classification / A. Patle, C. Deepak. *International Conference on Advances in Technology and Engineering (ICATE)* / A. Patle, C. Deepak., 2014.
45. Schmidhuber J. Deep learning in neural networks: An overview / Jurgen Schmidhuber. *Neural Networks* / Jurgen Schmidhuber., 2016. С. 85–117.
46. *Sensor-Based Activity Recognition* / [C. Liming, H. Jesse, C. Diane та ін.] // *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART C: APPLICATIONS AND REVIEWS* / [C. Liming, H. Jesse, C. Diane та ін.], 2013. С. 790–808.
47. Sokolova M. *AI 2007: Advances in Artificial Intelligence* / M. Sokolova, N. Japkowicz, S. Szpakowicz. Beyond Accuracy, F-score and ROC: a Family of
48. Steyvers M. Probabilistic Topic Models / M. Steyvers, T. Griffiths. *Handbook of latent semantic analysis* / M. Steyvers, T. Griffiths., 2008. С. 427–448.
49. Tapia E. Activity Recognition in the Home Using Simple and Ubiquitous Sensors / E. Tapia, I. Stephen, K. Larson. *International Conference on Pervasive Computing* / E. Tapia, I. Stephen, K. Larson, 2005. С. 158–175.
50. Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments / K. Humayun, R. Hoque, T. Keshav, S. Yang // *International*

- Journal of Distributed Sensor Networks / K. Humayun, R. Hoque, T. Keshav, S. Yang., 2017.
51. Vapnik V. Support-vector networks / V. Vapnik, C. Corinna. Machine learning / V. Vapnik, C. Corinna., 1995. С. 273–297.
52. Voulgaris Z. Extensions of the k nearest neighbour methods for classification problems / Z. Voulgaris, G. Magoulas. The 26th IASTED conference on artificial intelligence and applications / Z. Voulgaris, G. Magoulas., 2009. С. 23–28.
53. Yan Y. Hands-On Data Science with Anaconda / Y. Yan, J. Yan., 2019. 364 с.
54. Yuxuan L. Improving k Nearest Neighbor with Exemplar Generalization for Imbalanced Classification / L. Yuxuan, Z. Xiuzhen. Pacific-Asia Conference on Knowledge Discovery and Data Mining / L. Yuxuan, Z. Xiuzhen. С. 321–332.
55. WSU CASAS Datasets. URL: <http://casas.wsu.edu/datasets/> (дата звернення: 15.20.2021).
56. Smart home – Statistics & Facts. URL: <https://www.statista.com/topics/2430/smarthomes/> (дата звернення: 15.20.2021).
57. WSU CASAS Datasets URL: <http://casas.wsu.edu/datasets/> (дата звернення: 15.20.2021).
58. Serge T. User Activity Recognition in Smart Homes Using Pattern Clustering Applied to Temporal ANN Algorithm / T. Serge, Y. Yoo, M. Bourobou. *Open Access Journals Sensors*. 2016. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4481973/> (дата звернення: 15.20.2021).
59. Smart home – Statistics & Facts URL: <https://www.statista.com/topics/2430/smart-homes/> (дата звернення: 15.20.2022).

ДОДАТКИ**Додаток А****Лістинг програмного коду роботи з датчиками**

```

Main. c
#include «Board_LED. h»
#include «decode. h»
#include «i2c. h»
#include «ssd1306. h»
#include «ssd1306_fonts. h»
#include «stm32h7xx_hal. h»
#define RECEIVE_ADDRESS 0x0F
void RCC_Config (void);
int main (void)
{
    FILE · encodedIMG;
    char · decodedMsg;
    HAL_StatusTypeDef i2cStatus;
    uint8_t data;
    RCC_Config();
    if (I2C_Init() == HAL_ERROR || decoderInit() ==
HAL_ERROR)
    {
        while (1)
        {
            LED_SetOut (2U);
            HAL_Delay (400U);
        }
    }
    ssd1306_Init();
    ssd1306_Fill (Black);
    encodedIMG = fopen («img. jpg», «r»);
    while (i2cStatus!=HAL_ERROR || i2cStatus!=HAL_TIMEOUT)
    {
        i2cStatus = I2C_Read (RECEIVE_ADDRESS, data);
        LED_SetOut (1U);
        HAL_Delay (50U);
        fputc (data, encodedIMG);
    }
    decodedMsg = decodeMessage (encodedIMG);
    ssd1306_WriteString (decodedMsg, Font_7x10, White);
    while (1)
    {
        LED_SetOut (1U);
        HAL_Delay (200U);
    }
}
void RCC_Config (void)
{

```

```

    __HAL_PWR_VOLTAGESCALING_CONFIG
(PWR_REGULATOR_VOLTAGE_SCALE1);
    |RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.                ClockType                =
RCC_CLOCKTYPE_SYCLK|RCC_CLOCKTYPE_PCLK1
    HAL_NVIC_SetPriority (SysTick_IRQn, 0, 0);
    HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5);
    HAL_RCC_OscConfig(&RCC_OscInitStruct);
    HAL_SYSTICK_CLKSourceConfig (SYSTICK_CLKSOURCE_HCLK);
    HAL_SYSTICK_Config (HAL_RCC_GetHCLKFreq()/1000);
    RCC_ClkInitStruct. APB1CLKDivider = RCC_HCLK_DIV4;
    RCC_ClkInitStruct. AHBCLKDivider = RCC_SYCLK_DIV1;
    RCC_ClkInitStruct. APB2CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct. SYSCLKSource = RCC_SYCLKSOURCE_PLLCLK;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_OscInitStruct. HSEState = RCC_HSE_ON;
    RCC_OscInitStruct. OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct. PLL. PLLN = 336;
    RCC_OscInitStruct. PLL. PLLM = 8;
    RCC_OscInitStruct. PLL. PLLQ = 7;
    RCC_OscInitStruct. PLL. PLLState = RCC_PLL_ON;
    RCC_OscInitStruct. PLL. PLLP = RCC_PLL2DIVR_P2;
    RCC_OscInitStruct. PLL. PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitTypeDef RCC_OscInitStruct;
}

```